Development of a Conceptual Design Mission Analysis System for Guided Entry Systems

Space Systems Design Lab (SSDL)
Guggenheim School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA

June 30, 2008

Table of Contents

I. Planetary Entry System Synthesis Tool V0.1.1 User's Guide

A. PESST Overview

The Planetary Entry Systems and Synthesis Tool (PESST) is a rapid conceptual design tool developed by the Space Systems Design Laboratory (SSDL) at the Georgia Institute of Technology. This framework has the capability to estimate the performance and mass of an entry system using user-defined geometry, hypersonic aerodynamics, flight mechanics, thermal response, and mass estimation. Earth and Mars atmospheres are preloaded with the ability to also use either user-defined or GRAM atmospheric models. Trade studies can be performed by parameter sweeps to gain an excellent understanding of the design space for conceptual studies. This framework is broadly applicable to conceptual studies of EDL systems with varying landed precision requirements.

B. Running PESST

The tool may be run from either the command line or through a Java graphical user interface (GUI).

1. Starting the Graphical User Interface

1) Change to the directory that contains the pesst.jar executable jar file.
2) Run the command, '**java –jar pesst.jar**'

The GUI will open and allow one to load or execute cases. Use the help button within the GUI for information regarding the different input options.

2. Using the Command Line

When the user clicks on 'Calculate' in the graphical user interface, the file called examplePesst.in is created. This is the file format used by the command line tool that is run by the GUI. The user can run this file directly from the command line; this can be useful as the GUI caches the output until it has a full buffer before printing to the graphical screen while, from the command line, the output is shown as it is created.

The two executables that make up PESST are called '**generateAero**' and '**runPesst**'. Executing either without arguments will display a listing of the inputs required by each.

3. Geometry Utility Scripts

Three utility scripts have been added to assist the user with arbitrary geometries. PESST currently accepts .gts geometry files.

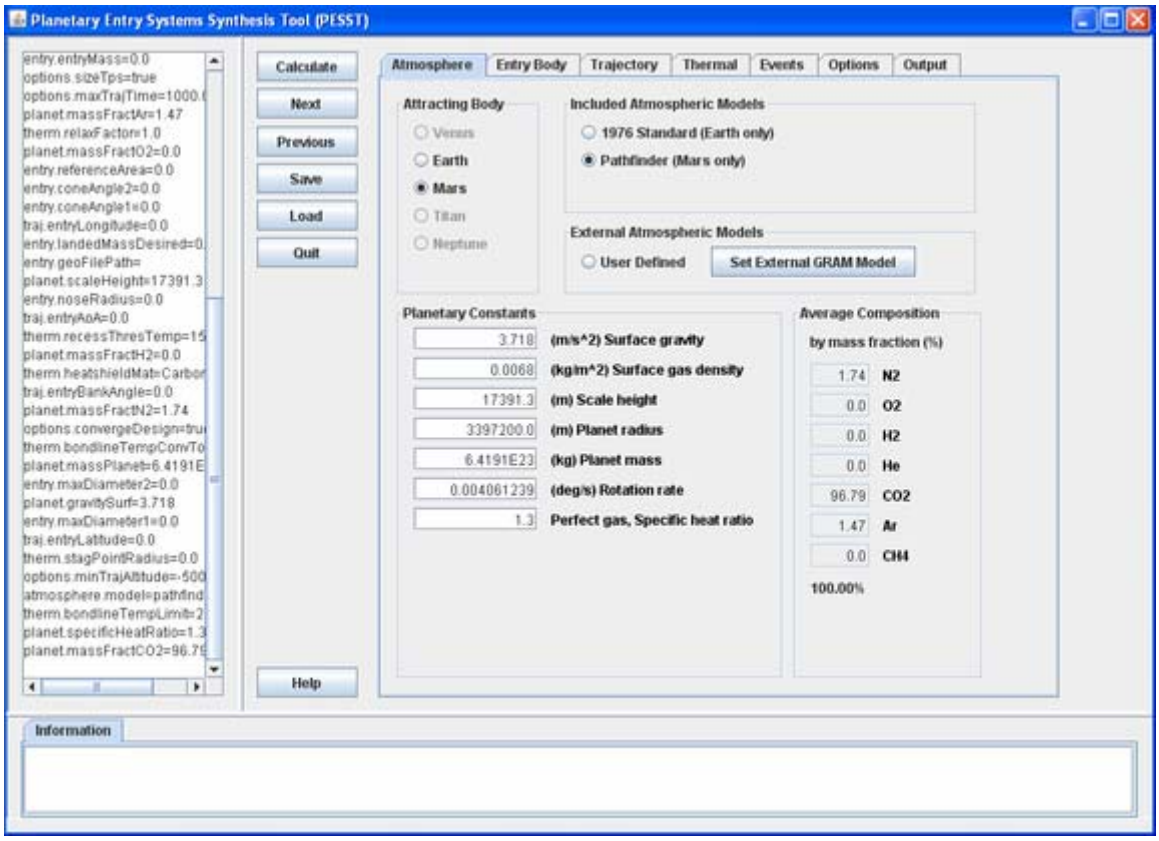
To convert a common stl file to a gts file:
stl2gts < stl file > gts file

To convert a gts file to a stl file:
gts2stl < gts file > stl file

To check the correctness of a gts file
gtscheck < gts file

C. PESST GUI

1. Atmospheric Tab



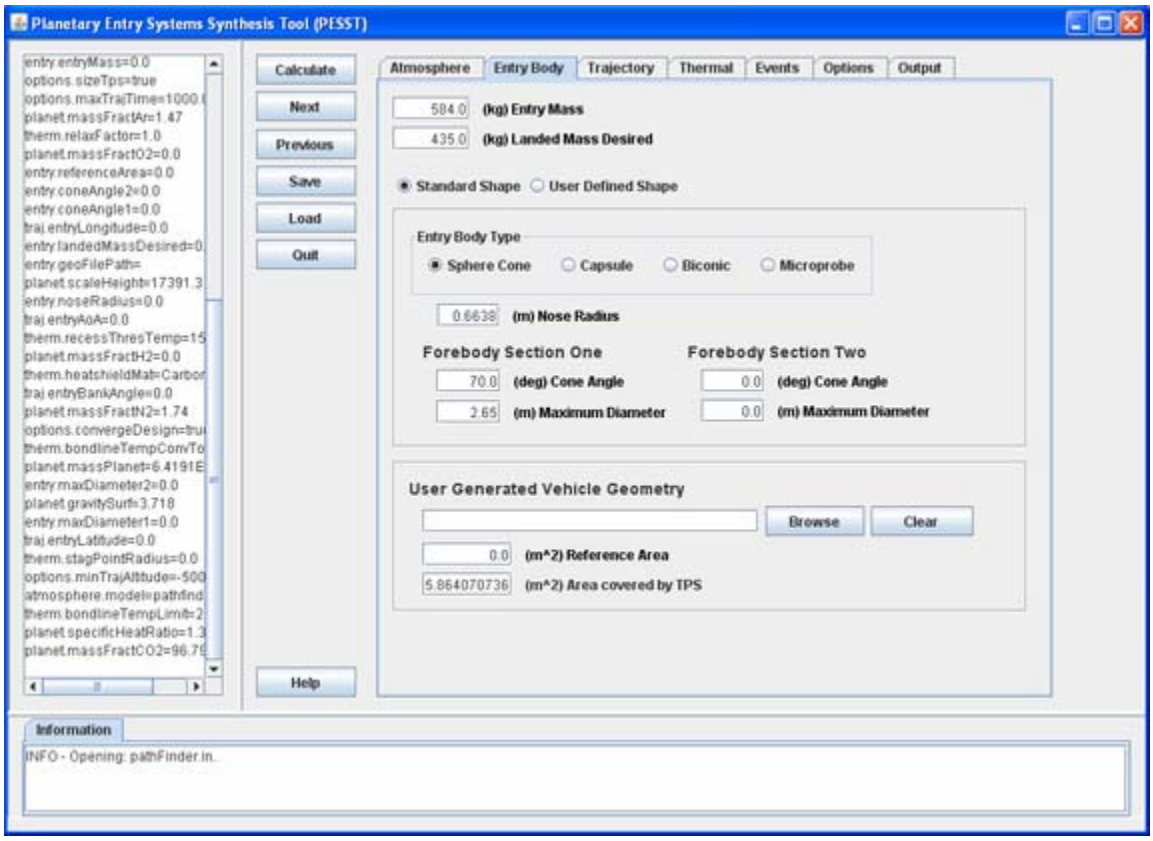
Attracting Body:

- The user can select the main attractive body. Planet rotation rate, gravity, Sutton-Graves constant, etc. are set by this choice.
- Earth and Mars supported through heating models

Atmospheric Models:

- Earth—Standard 1976 Earth Atmosphere.
- Mars—Reconstrcuted atmosphere from the Mars Pathfinder mission.

2. Entry Body Tab



Entry Mass:

- Initial guess for the total entry mass of the vehicle. This is provided by the user as an initial guess that will be converged onto a final entry mass.

Landed Mass Desired:

- The mass the user wishes to arrive at the surface. This value will not be changed during the convergence loop.

a) Using Standard Shapes

Entry Body Type:

- The user can currently select between four standard body types; Sphere Cone, Capsule, Biconic, Microprobe. The user only needs to give geometric information regarding the forward face of the entry vehicle as Newtonian (or modified Newtonian) aerodynamics only applies to the windward surface.

Nose Radius:

- The radius of the spherical segment at the forward most tip of most vehicles. This value is required for all the stock shapes.

Section One:

- Parameters used to define the forward most section of the body (a biconic will have two sections)

Section Two:
- Required to specify the parameters of the second condical segment of a biconic.

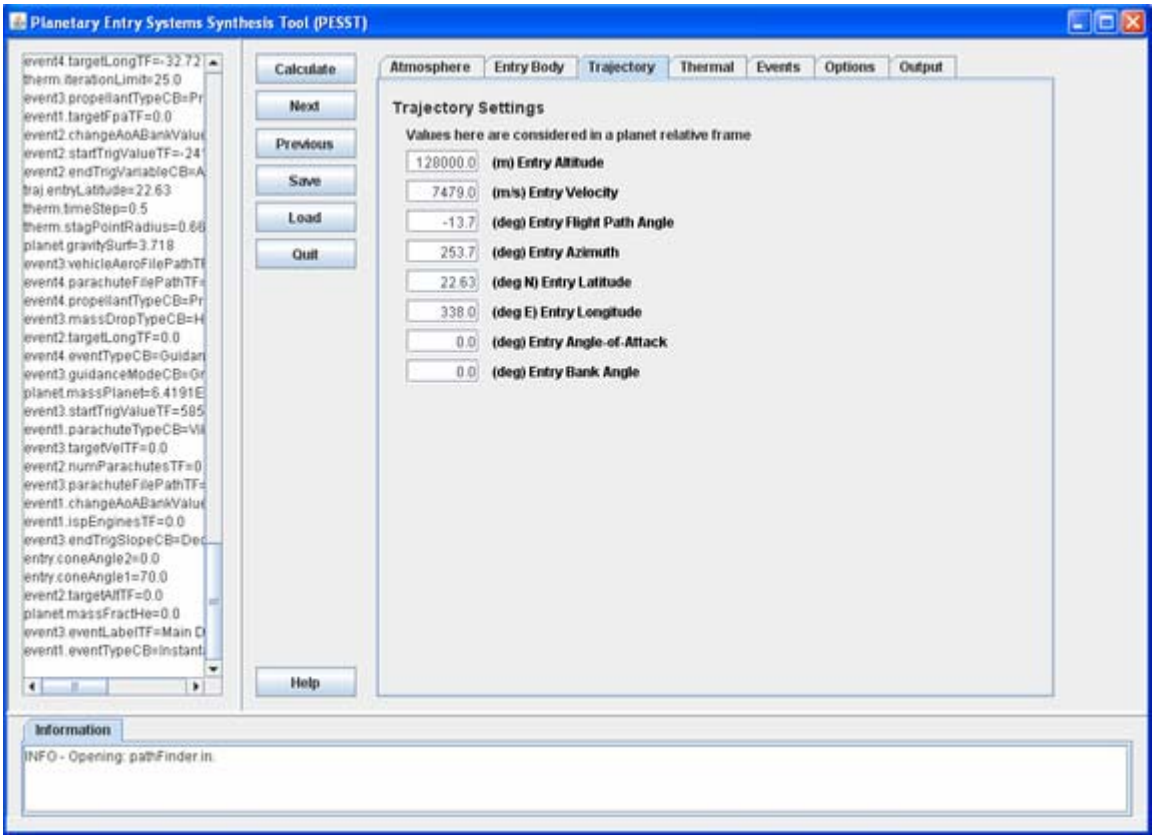
Cone Angle :
- The half angle for the cone segment.

Maximum Diameter :
- The maximum diameter reached by this conical segment.

b) Using User Define Shapes

- **Path to .gts geometry file**:
    - There is a utility function called stl2gts.exe to help convert a common .stl file to a .gts file.
    - The forward body axis of the vehicle must be along the positive x-axis of the geometry file.
    - Changes to the angle of attack will occur by rotating the vehicle about the y-axis.
    - All panel normal vectors must be pointing outwards. This can be checked graphically by using the freely available mview tool. This tool also has the ability to toggle the direction of the normal vectors if they happen to be pointing inwards.
    - An example geometry file can be found under the examples directory.
    - **Note for Windows GUI users**: The path may not have spaces in it or the geometry file will not be found. This restriction is only for GUI users; Windows users using PESST from the command line may use paths with spaces by using quotes around the path passed in to the generateAero executable.
- **Reference area**:
    - Area used to normalize the aerodynamic forces
- **Area covered by TPS**:
    - The option allows the user to manually specify how much area is covered by the TPS. This TPS is sized by the heating experienced at the stagnation point and is uniformly applied over the coverage area.

3. Trajectory Tab



Entry Altitude:
- Assumes spherical planet radius.

Entry Velocity:
- Planet relative.

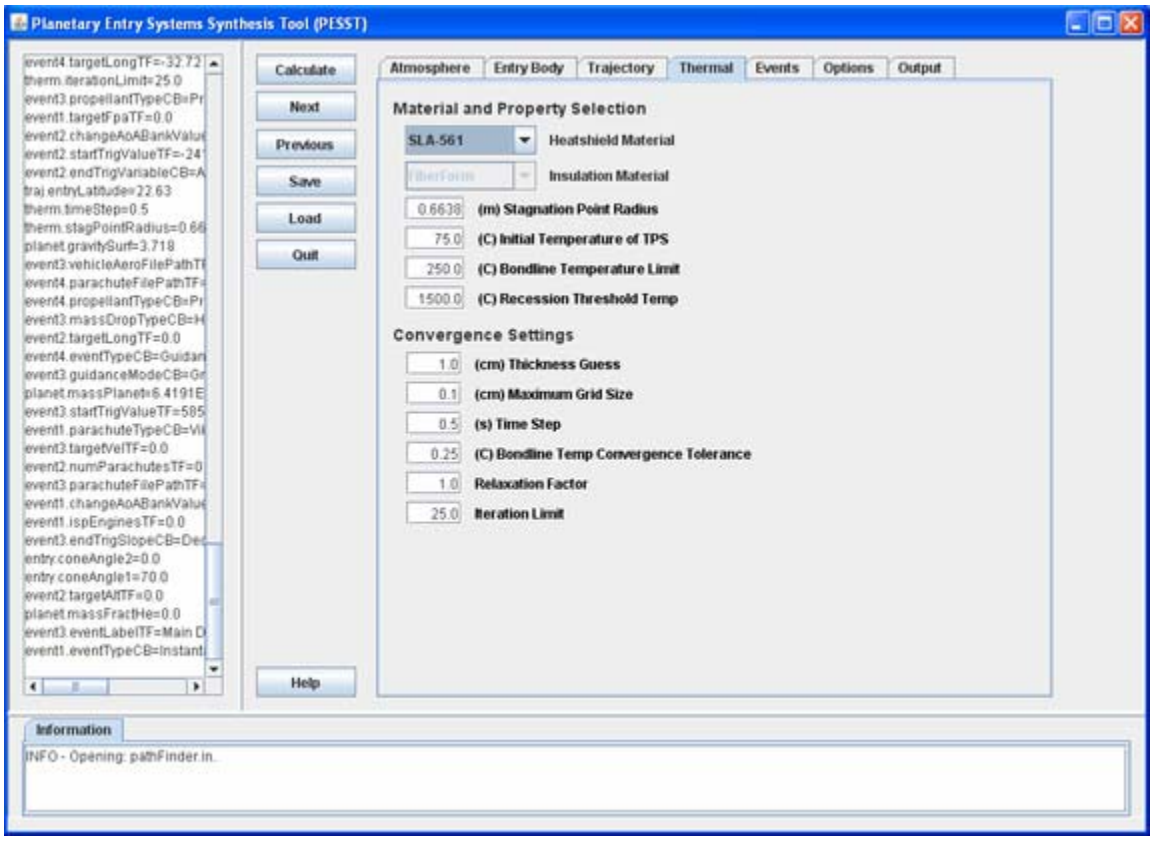
Entry Flight Path Angle:
- Positive up convention.

Entry Azimuth:
- Angle from North, positive clockwise.

Entry Angle-of-Attack:
- Range from -45 to 45 deg computed, used for lifting entries.

Entry Bank Angle:
- Initial bank angle.

4. Thermal Tab



Stagnation Point Radius:

- This radius is used in the Sutton-Graves heating relation for the stagnation point heating.

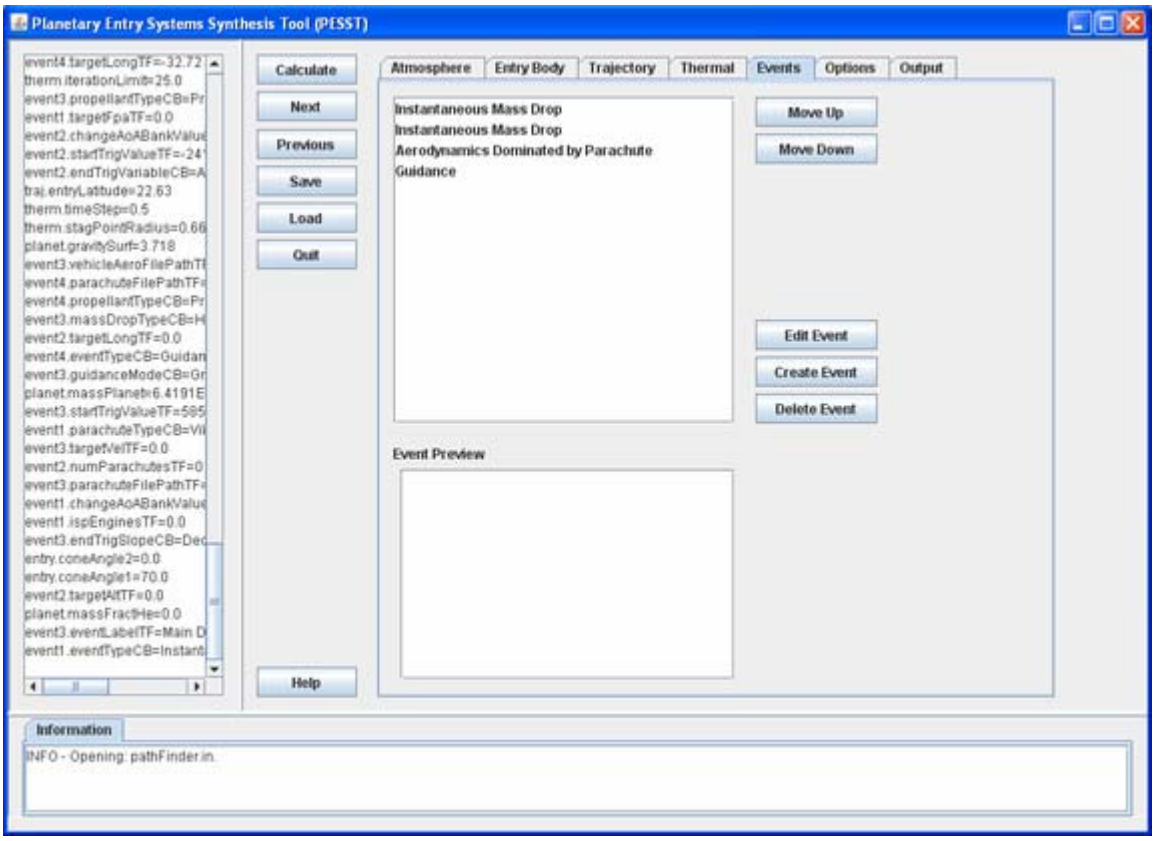
Bondline Temperature Limit:

- The maximum allowed temperature for the bondline. This is used to size the virgin TPS material required.

Recession Threshold Temp:

- Temperature above which recession will occur. TPS material interweaved with glass would likely have a different recession threshold temp compared with carbon-carbon TPS.

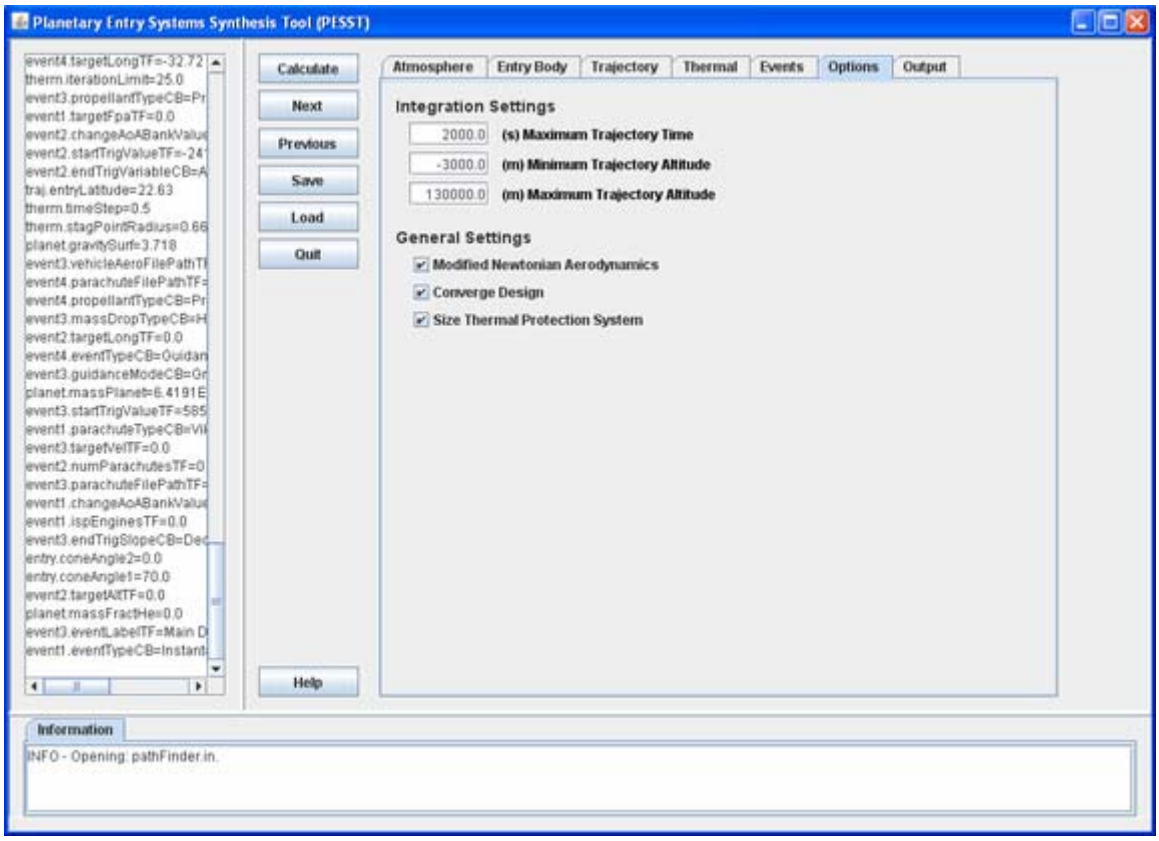
5. Event Tab



Create Event:

- Allows a user to create either an instantaneous event, such as a heatshield drop, or a event that lasts over a period of time such as guidance or a deployed parachute.
- Trigger values are used to signal when an event should begin or end
- The ordering of events in the window does not effect the order of their execution. Only the trigger targets matter in turning on or off an event.

6. Options Tab



Maximum Trajectory Time:

- If the trajectory surpasses this time limit; the integrator will be stopped.

Minimum Trajectory Altitude:

- If the trajectory travels below this altitude; the integration will be stopped.

Maximum Trajectory Altitude:

- If the trajectory travels above this altitude; the integration will be stopped.

Modified Newtonian Aerodynamics:

- Turn off for classical Newtonian aerodynamics. Turn on to use modified Newtonian aerodynamics.

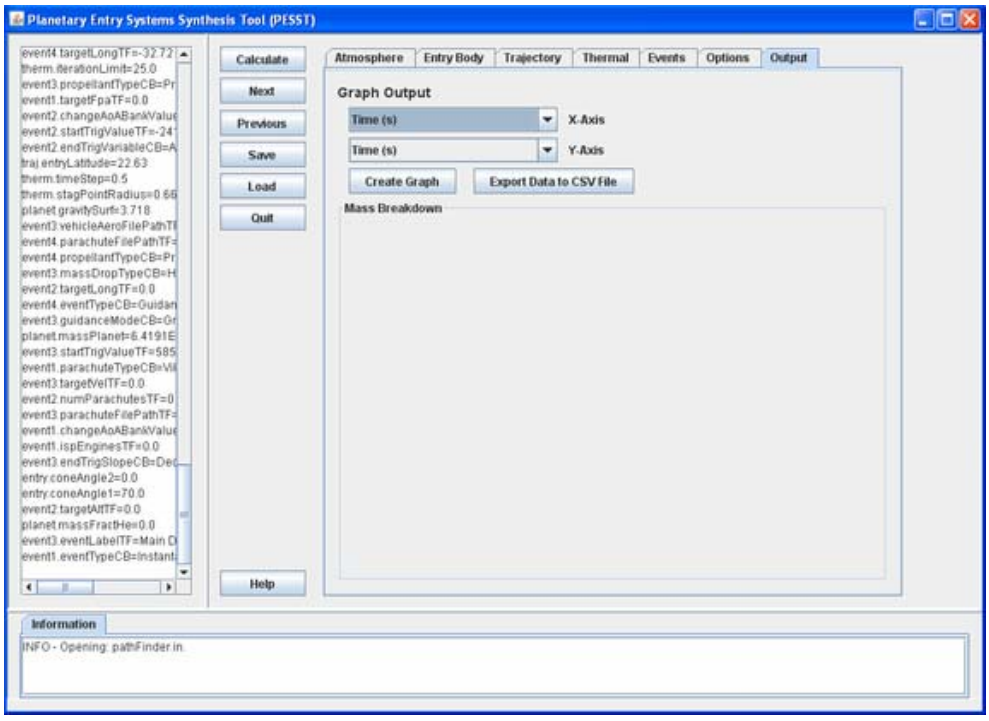
Converge Design:

- Turn off to only pass once through the disciplines. Turn on to converge design mass.

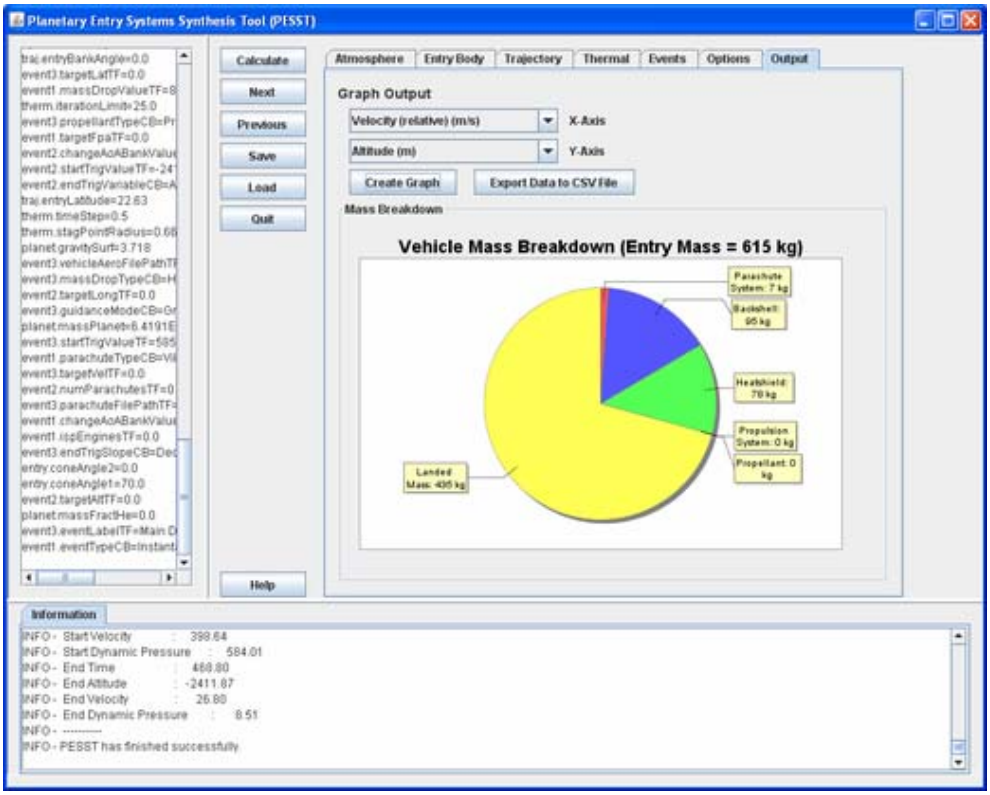
Size Thermal Protection System:

- This will size the TPS system required for the heating environment.

7. Output Tab



Data will only be displayed here once a case has been run. The user can generate graphs on the returned data or output it to a .csv file.



D. PESST Computational Modules

PESST has six primary modules—planetary model, system aerodynamics, guidance, navigation, and control, trajectory analysis, thermal response, and weights and sizing. Each of these modules is responsible for a separate component of the analysis and interact with one another. A design structure matrix (DSM) describing this interaction is shown in Figure 1, with dots representing the transferring of a data from one module to another.

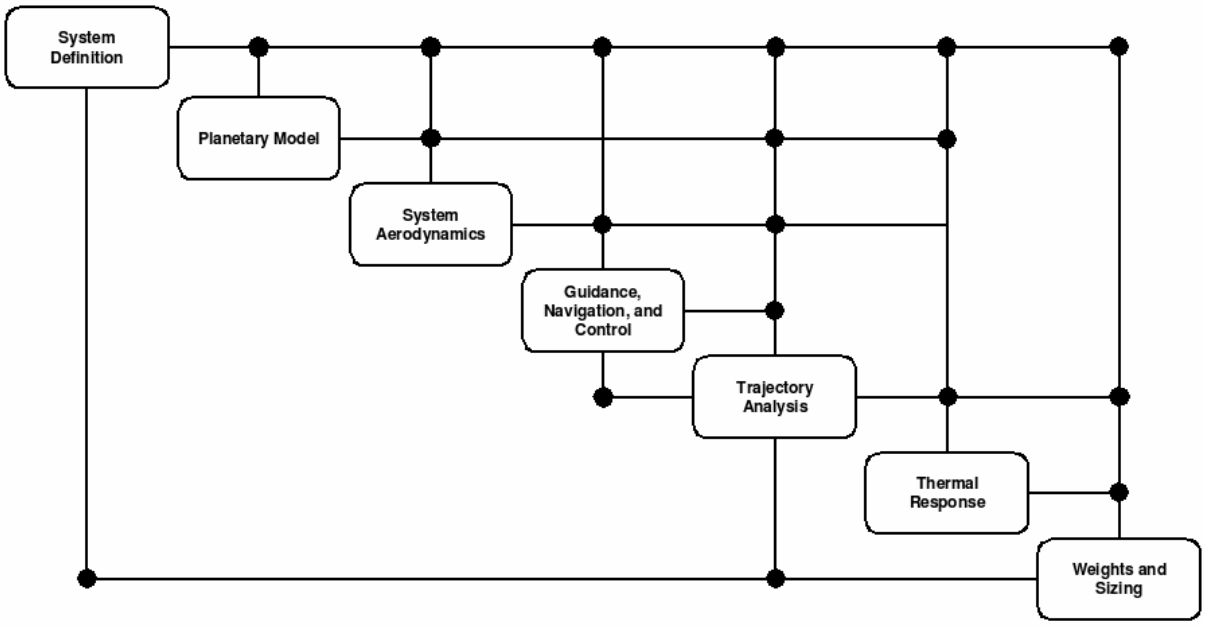


Figure 1: Design structure matrix for PESST.

1. System Definition

The purpose of this module is to define the system parameters and options.

Specified by the module:

- For standard vehicle shapes; the user would select from the current built in geometries
- Selected planet
- Selected initial conditions
- Selected TPS materials
- Events that will occur during the entry sequence
- Initial guess for entry mass and the desired landed payload mass

2. System Aerodynamics

The user has the choice of either using classical or modified Newtonian aerodynamics to characterize the vehicle's aerodynamic performance. This method estimates the aerodynamics of a flat plate exposed to hypersonic flow. By paneling a vehicle's surface, treating each panel individually and adding the accumulated forces together; the hypersonic aerodynamics for arbitrary shapes can be approximated.

a) Mesh Creation

The mesh of panels for both the standard and arbitrary bodies is generated by the open-sourced GNU Triangulated Surface Library. This LGPL Library can generate a mesh from a descriptive equation or from an input geometry file. Standard entry shapes are described by equations that are sampled by the library for the vertices required for the mesh.

b) Standard Shapes

The equations for both a sphere and a conical frustum are combined to provide the library with the vertex points needed to describe the surface of a sphere-cone, see Figure 2.

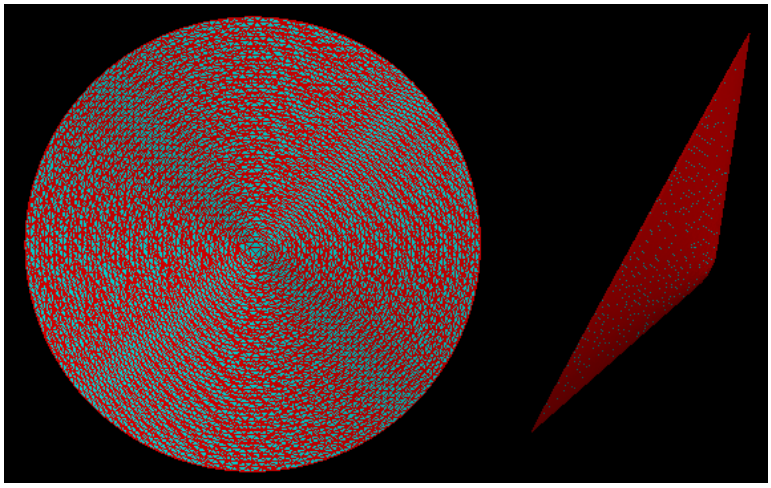


Figure 2: 70 degree sphere-cone shape.

The equation for an additional conical frustum is used to describe a biconic entry vehicle shape, Figure 3. The same principle could be used for triconic shapes in the future. The triangles in the formed meshes are treated as individual panels for the Newtonian aerodynamic approximation.

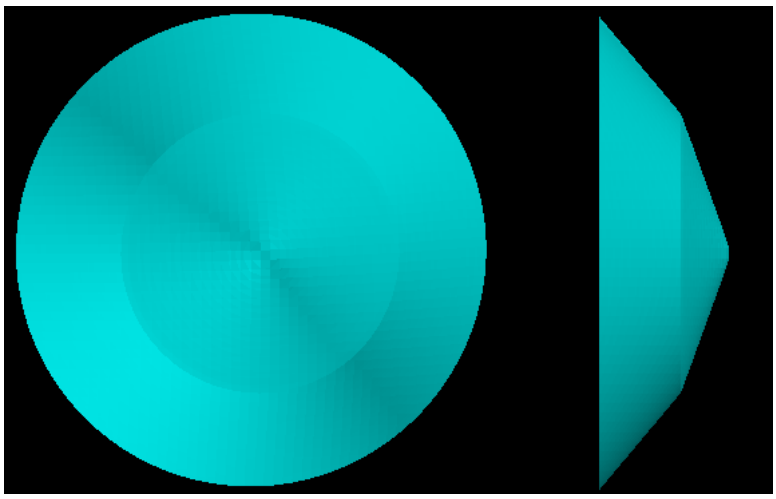


Figure 3: 70/60 degree biconic shape.

c) Using Arbitrary Geometries

The user may also examine geometries that they define through the use of their own CAD packages. User defined geometries must conform to a few requirements to be compatible with the PESST framework.

1) As can be seen in Figure 4, the positive x-axis must align with the forward body axis of the vehicle.
2) The vehicle will be rotated about the y-axis for different angles of attack. The user defined geometry should keep this in mind.

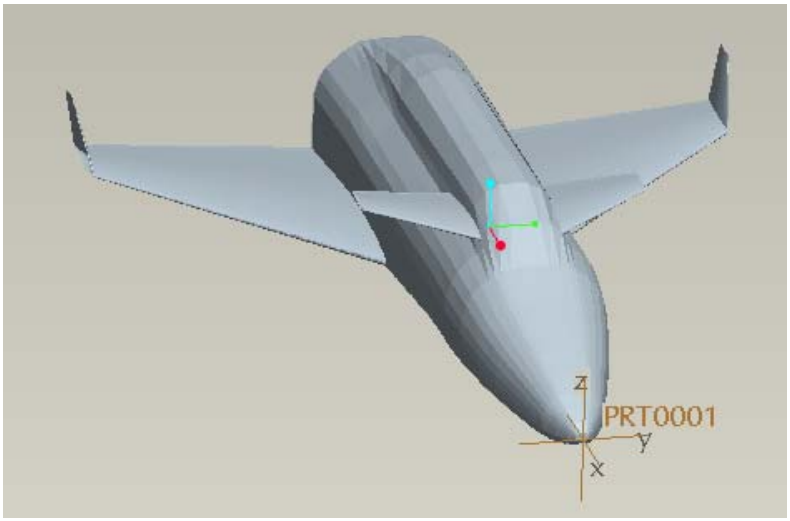


Figure 4: STL file for arbitrary geometry using ProEngineer.

3) PESST takes GTS files as inputs. To convert from a common STL file, utility scripts have been included. The mview tool is extremely useful for visualizing the geometry.

```
stl2gts.exe < exampleGeo.stl > exampleGeo.gts
mview.exe < exampleGeo.gts
```

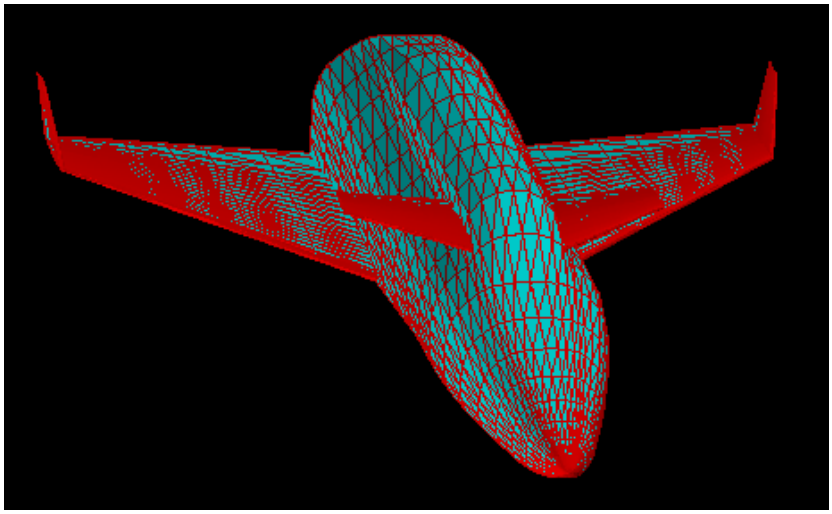


Figure 5: GTS file converted from stl geometry file.

4) The normal vectors for the shape being used with PESST must point outwards for the aerodynamic code to properly determine the windward panels of the vehicle. Figure 6 shows an example of a GTS file with its normal vectors pointing in the wrong direction. The normal direction must be checked.

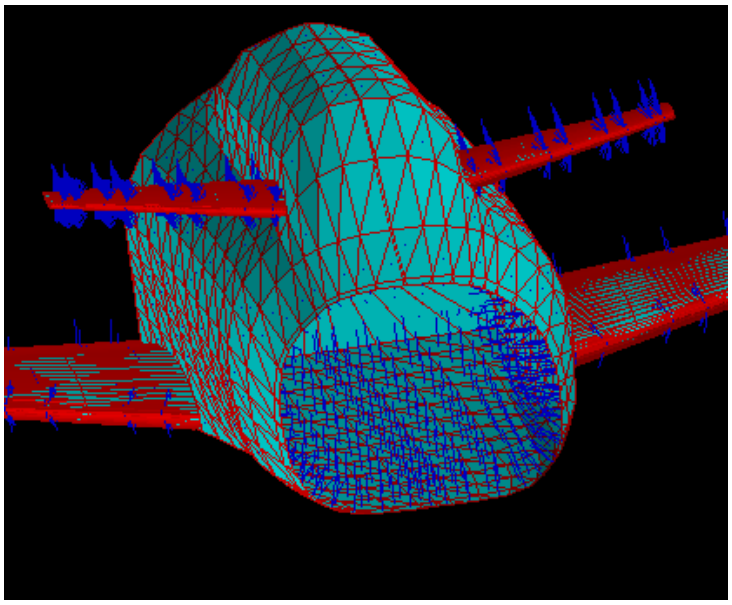


Figure 6: Example of inward facing normal vectors.

5) If the normal vectors are pointing inwards, use the freely available mview tool to flip the normal vectors, Figure 7. Save the GTS file with the flipped normal vectors. Now it is ready to be input into PESST.

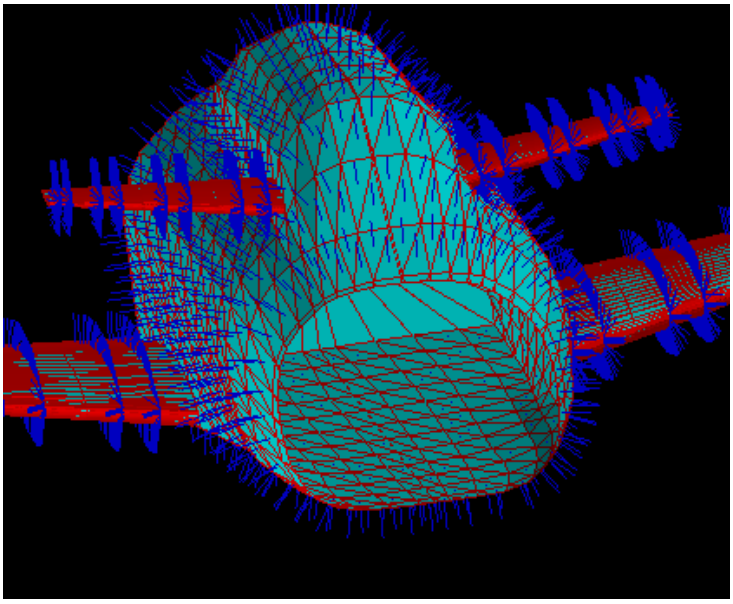


Figure 7: Example of outward facing normal vectors.

3. Guidance, Navigation, and Control

a) Gravity Turn

A gravity turn is a propulsive terminal descent guidance law that applies thrust to directly oppose the vehicle's velocity, as shown in Figure 8.

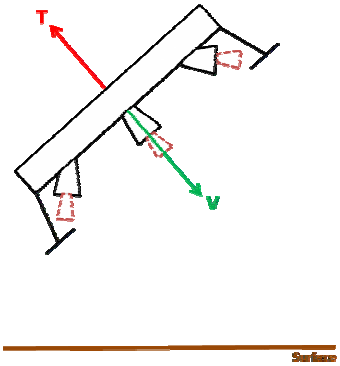


Figure 8: Gravity turn schematic.

Throughout the gravity turn maneuver, the thrust magnitude is constant and can be computed by several methods. An analytic thrust magnitude solution is able to be obtained if a small nadir angle is assumed. However, for generality, PESST implements the gravity turn by propagating the equations of motion described in the Trajectory Analysis section using a fixed thrust magnitude and determining the end state and uses a Newton iteration scheme to converge on the desired final altitude and velocity magnitude. As mentioned before, the thrust magnitude found through this iteration is theoretically constant; however, unknown disturbances in the system can cause the dynamics of the system to vary along with the required thrust magnitude and therefore the gravity turn guidance law is implemented in a closed-loop sense to ensure the desired terminal altitude and velocity is achieved.

b) Using Gravity Turn

The gravity turn guidance law is found within PESST in the ***Events*** tab of the GUI and can be created by:

1. Selecting *Create Event* from the ***Events*** tab of the GUI
2. Selecting *Guidance* from the ***Event Type*** drop-down menu
3. Selecting the appropriate initiation values for the guidance event in the ***Start trigger for Event*** field (*e.g.*, initiating the propulsive terminal descent at 8 km altitude when the slope is decreasing requires selecting *Altitude (m)* from the ***Trigger Variable*** drop-down menu, selecting *Decreasing* from the ***Trigger Slope*** drop-down menu, and typing *8000* into the ***Trigger Value*** field)
4. Selecting the appropriate termination values for the guidance event in the ***End trigger for Event*** field (*e.g.*, terminating the propulsive terminal descent at 0 km altitude when the slope is decreasing requires selecting *Altitude (m)* from the ***Trigger Variable*** drop-down menu, selecting *Decreasing* from the ***Trigger Slope*** drop-down menu, and typing *0* into the ***Trigger Value*** field)

5. Inputting a name for the guidance event in the Event Label field (*e.g.*, typing *Gravity Turn* into that field)
6. Within the ***Guidance*** field:
    a. Selecting *Gravity Turn* from the ***Guidance*** pull-down menu
    b. Inputting the target altitude in meters in the ***(m) Target Altitude*** text field
    c. Inputting the target velocity in meters per second in the ***(m/s) Target Velocity*** text field
    d. In the Engine Information section, identifying the parameters of the terminal descent engines
        i. Selecting the propellant type from the ***Propellant Type*** pull-down menu
        ii. Selecting the propellant tank material from the ***Propellant Tank Material*** pull-down menu
        iii. Inputting the specific impulse (in seconds) of the engines by typing the value into the ***(s) Specific Impulse of Engines*** field
        iv. Inputting the maximum thrust (in Newtons) for all of the engines by typing the value into the ***(N) Max Thrust from Engines*** field

    NOTE: the remaining fields do not concern the gravity turn guidance law and will not influence its performance.
7. Selecting the ***OK*** button in the bottom right hand corner of the window

An example of a completed event window configured for the gravity turn is shown in Figure 9.

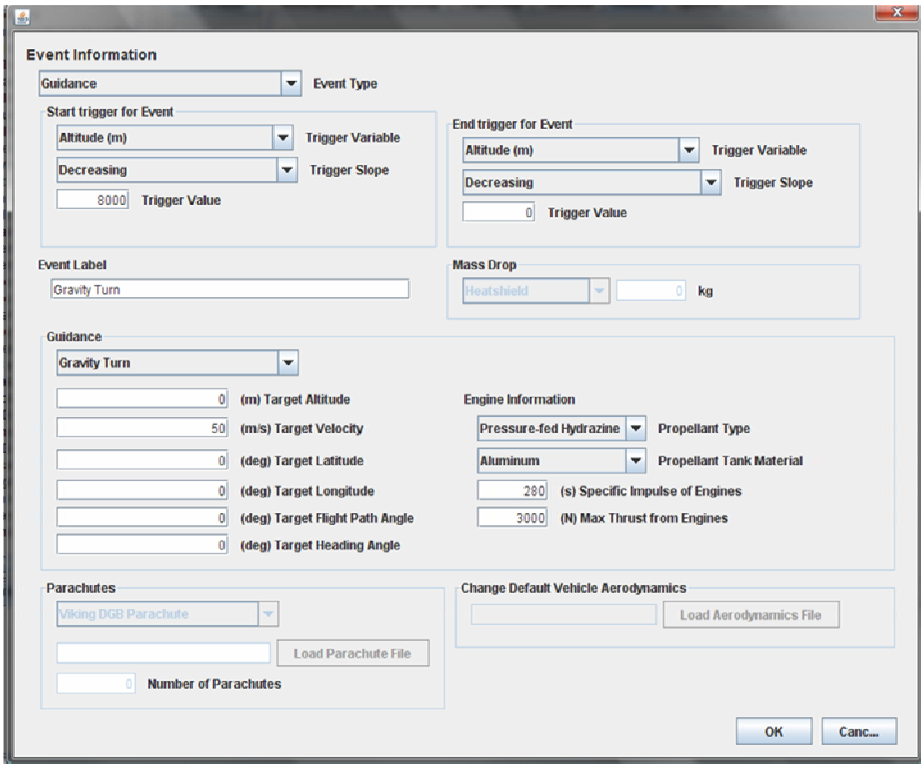


Figure 9: PESST event window with the gravity turn guidance law configured.

c) Closed-Form Analytic Method Theory

By assuming a planar, non-rotating planet with no atmosphere, D'Souza derived an analytic, unconstrained fuel-optimal propulsive terminal descent algorithm that meets the necessary and sufficient conditions for an optimal control law. The problem described by D'Souza minimizes the performance index

$$J = \Gamma t_f + \frac{1}{2}\int_{t_0}^{t_f} \mathbf{a}^T \mathbf{a} dt \tag{1}$$

which includes a weighting, Γ, on the final time (assumed to be zero for the implementation within PESST). It is shown that the control law which minimizes this index, under the assumptions mentioned previously, is given by

$$\mathbf{a} = -4\frac{\Delta\mathbf{v}}{t_{go}} - 6\frac{\Delta\mathbf{r}}{t_{go}^2} - \mathbf{g} \tag{2}$$

where

$$\Delta\mathbf{r} = \begin{pmatrix} r_1 - r_{f_1} & r_2 - r_{f_2} & r_3 - r_{f_3} \end{pmatrix}^T \tag{3}$$

$$\Delta\mathbf{v} = \begin{pmatrix} v_1 - v_{f_1} & v_2 - v_{f_2} & v_3 - v_{f_3} \end{pmatrix}^T \tag{4}$$

$$\mathbf{g} = \begin{pmatrix} 0 & 0 & g \end{pmatrix}^T \tag{5}$$

The time-to-go, t_{go}, is then the real positive root of the equation

$$t_{go}^4 - 2\frac{\Delta\mathbf{v}^T\Delta\mathbf{v}}{g^2/2}t_{go}^2 - 12\frac{\Delta\mathbf{v}^T\Delta\mathbf{r}}{g^2/2}t_{go} - 18\frac{\Delta\mathbf{r}^T\Delta\mathbf{r}}{g^2/2} = 0 \tag{6}$$

In PESST, Eq. (6) is solved using a Newton iteration, which provides sufficiently fast convergence and eliminates much of the numerical issues associated with the analytical root equations. The commanded thrust is then

$$\boldsymbol{\tau} = m\left(-4\frac{\Delta\mathbf{v}}{t_{go}} - 6\frac{\Delta\mathbf{r}}{t_{go}^2} - \mathbf{g}\right) \tag{7}$$

Within PESST, the propulsive terminal descent solution, Eq. (7) is computed in a closed-loop sense, being called at each propagated timestep of the system equations of motion after the selected trigger is activated.

d) Using the Closed-Form Analytic Method

The closed-form analytic guidance law is found within PESST in the ***Events*** tab of the GUI and can be created by:

1. Selecting *Create Event* from the ***Events*** tab of the GUI
2. Selecting *Guidance* from the ***Event Type*** drop-down menu
3. Selecting the appropriate initiation values for the guidance event in the ***Start trigger for Event*** field (*e.g.*, initiating the propulsive terminal descent at 8 km altitude when the slope is decreasing requires selecting *Altitude (m)* from the ***Trigger Variable*** drop-down menu, selecting *Decreasing* from the ***Trigger Slope*** drop-down menu, and typing *8000* into the ***Trigger Value*** field)
4. Selecting the appropriate termination values for the guidance event in the ***End trigger for Event*** field (*e.g.*, terminating the propulsive terminal descent at 0 km altitude when the slope is decreasing requires selecting *Altitude (m)* from the ***Trigger Variable*** drop-down menu, selecting *Decreasing* from the ***Trigger Slope*** drop-down menu, and typing *0* into the ***Trigger Value*** field)
5. Inputting a name for the guidance event in the Event Label field (*e.g.*, typing *Guided Propulsive Terminal Descent* into that field)
6. Within the ***Guidance*** field:
    a. Selecting *Analytic Terminal Descent—D'Souza* from the ***Guidance*** pull-down menu
    b. Inputting the target altitude in meters in the ***(m) Target Altitude*** text field
    c. Inputting the target velocity in meters per second in the ***(m/s) Target Velocity*** text field
    d. Inputting the target latitude in degrees in the ***(deg) Target Latitude*** text field
    e. Inputting the target longitude in degrees in the ***(deg) Target Longitude*** text field
    f. Inputting the target flight path angles in degrees the ***(deg) Target Flight Path Angle*** text field
    g. Inputting the target heading angle in degrees the ***(deg) Target Heading Angle*** text field
    h. In the Engine Information section, identifying the parameters of the terminal descent engines
        i. Selecting the propellant type from the ***Propellant Type*** pull-down menu
        ii. Selecting the propellant tank material from the ***Propellant Tank Material*** pull-down menu
        iii. Inputting the specific impulse (in seconds) of the engines by typing the value into the ***(s) Specific Impulse of Engines*** field
        iv. Inputting the maximum thrust (in Newtons) for all of the engines by typing the value into the ***(N) Max Thrust from Engines*** field
7. Selecting the ***OK*** button in the bottom right hand corner of the window

An example of a completed event window configured for the closed-form analytic guidance law is shown in Figure 10.

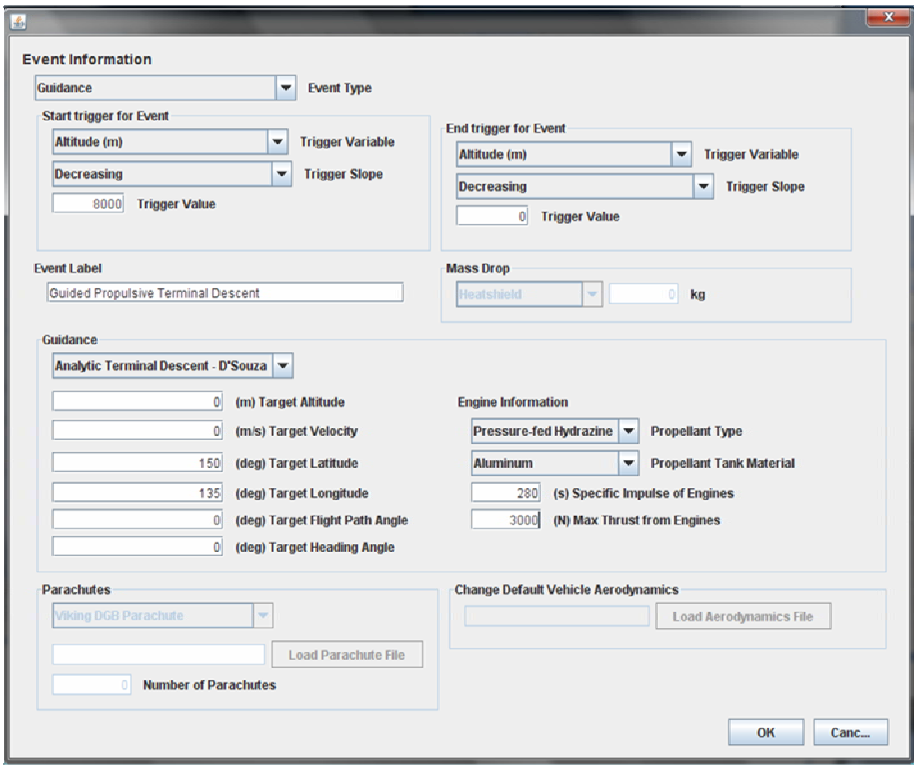


Figure 10: PESST event window with the closed-form analytic guidance law configured.

4. Trajectory Analysis

PESST utilizes three degree-of-freedom (DOF) with bank angle modulation equations of motion to determine the time history of the entry system's state. The trajectory is propagated using a variable-step 4^{th}-order Runge-Kutta integrator with 5^{th}-order error truncation over a fixed time-span from a set of planet relative initial conditions (altitude, velocity, flight path angle, azimuth angle, latitude, longitude, angle of attack, and bank angle) until the terminal altitude condition is achieved (either a maximum altitude or minimum altitude) or the propagation has exceeded an allotted time and has not reached the terminal altitude condition. These values are set in two different tabs of the GUI—the ***Trajectory*** tab and the ***Options*** tab as shown in Figure 11 and Figure 12, respectively.

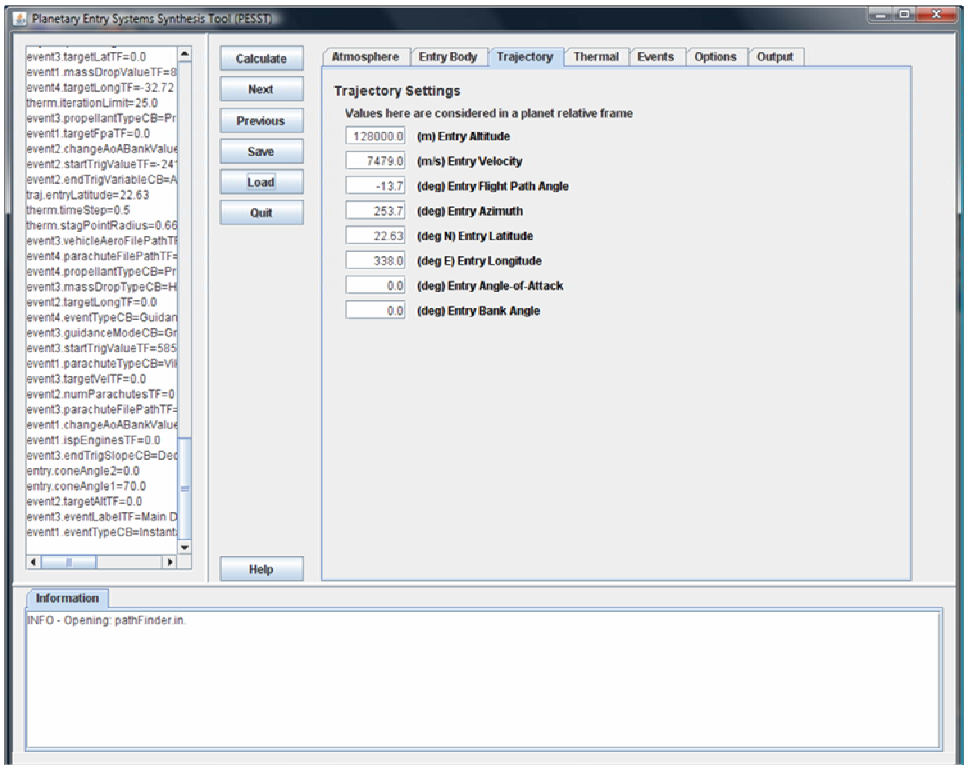


Figure 11: PESST trajectory window.

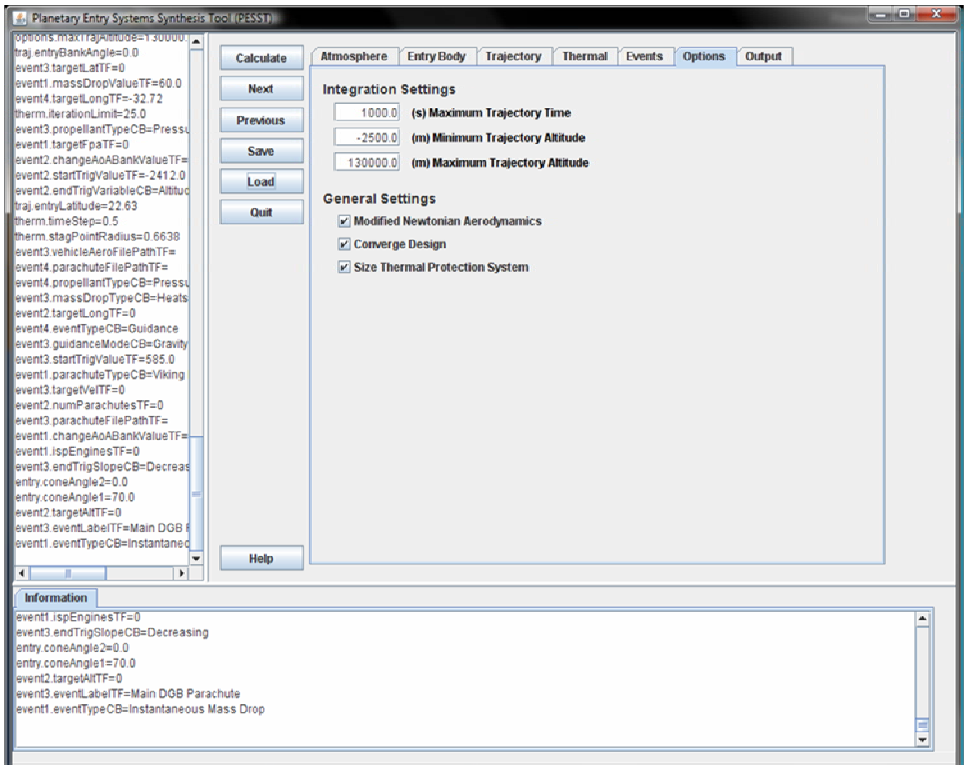


Figure 12: PESST options window.

The equations of motion for the system that are utilized are expressed in terms of their Cartesian coordinates as to eliminate singularities that can arise with the use of angles. The system dynamics are described by the relation

$$\begin{pmatrix} \dot{\mathbf{r}}^{\mathbf{I}} \\ \dot{\mathbf{v}}^{\mathbf{I}} \\ \dot{m} \end{pmatrix} = \begin{pmatrix} \mathbf{v}^{\mathbf{I}} \\ \sum \mathbf{f}^{\mathbf{I}} / m \\ f(T, I_{SP}) \end{pmatrix} \tag{8}$$

where:

$\mathbf{r}^{\mathbf{I}}$ = position vector from the center of the inertial frame
$\mathbf{v}^{\mathbf{I}}$ = velocity vector relative to the inertial frame
$\mathbf{f}^{\mathbf{I}}$ = force vector expressed in the inertial frame
m = mass of the entry system
T = thrust magnitude
Isp= specific impulse

In Eq.(8), the forces are expressed in the inertial frame and the derivatives are taken with respect to the inertial frame. Therefore, all of the forces acting on the entry system need to be expressed in the inertial frame. For instance, the force due to drag is most easily expressed in the body-frame and thus needs transformation into the inertial frame. This is done using a transformation matrix approach seen in Eq. (9) where L_{BA} is the transformation matrix from Frame A to Frame B, $\boldsymbol{\xi}^{\mathbf{A}}$ is an arbitrary vector expressed in Frame A, and $\boldsymbol{\xi}^{\mathbf{B}}$ is the same arbitrary vector expressed in Frame B.

$$\boldsymbol{\xi}^{\mathbf{B}} = L_{BA} \boldsymbol{\xi}^{\mathbf{A}} \tag{9}$$

The transformation matrix L_{BA} is given by

$$L_{BA} = \begin{pmatrix} \mathbf{i}^{\mathbf{B}} \cdot \mathbf{i}^{\mathbf{A}} & \mathbf{i}^{\mathbf{B}} \cdot \mathbf{j}^{\mathbf{A}} & \mathbf{i}^{\mathbf{B}} \cdot \mathbf{k}^{\mathbf{A}} \\ \mathbf{j}^{\mathbf{B}} \cdot \mathbf{i}^{\mathbf{A}} & \mathbf{j}^{\mathbf{B}} \cdot \mathbf{j}^{\mathbf{A}} & \mathbf{j}^{\mathbf{B}} \cdot \mathbf{k}^{\mathbf{A}} \\ \mathbf{k}^{\mathbf{B}} \cdot \mathbf{i}^{\mathbf{A}} & \mathbf{k}^{\mathbf{B}} \cdot \mathbf{j}^{\mathbf{A}} & \mathbf{k}^{\mathbf{B}} \cdot \mathbf{k}^{\mathbf{A}} \end{pmatrix} \tag{10}$$

where it is assumed that the vectors in Frame A have the form $\boldsymbol{\xi}^{\mathbf{A}}=\xi_x\mathbf{i}^{\mathbf{A}}+ \xi_y\mathbf{j}^{\mathbf{A}} +\xi_z\mathbf{k}^{\mathbf{A}}$ and the vectors in Frame B have the form $\boldsymbol{\xi}^{\mathbf{B}}=\xi_x\mathbf{i}^{\mathbf{B}}+ \xi_y\mathbf{j}^{\mathbf{B}} +\xi_z\mathbf{k}^{\mathbf{B}}$.

A free-body diagram describing the forces acting on the entry vehicle is shown in Figure 13. Notice the four primary forces acting on the body, weight, **W**, lift, **L**, thrust, **T**, and drag, **D**. The lift vector is dashed as it is shown as the projection into the plane shown, with an additional component being out-of-plane.

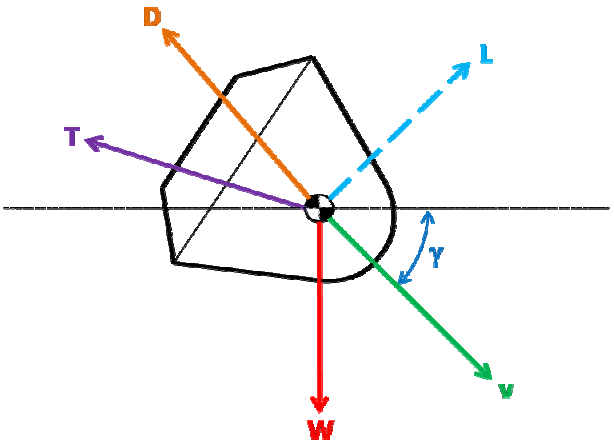


Figure 13: Entry system free-body diagram.

The weight vector, **W**, is expressed as

$$\mathbf{W} = m\mathbf{g} \tag{11}$$

The lift vector's magnitude is given by

$$L = -qC_L S \tag{12}$$

where:

q = dynamic pressure = $0.5\rho|\mathbf{v}^{\mathbf{Rel}}|^2$
C_L = coefficient of lift, discussed in the System Aerodynamics section
S = drag area of the entry system, discussed in the System Aerodynamics section

The lift vector is perpendicular to the velocity vector and its orientation relative to the plane shown in the free-body diagram (Figure 13) can be expressed in terms of the bank angle, σ, as shown in Figure 14.

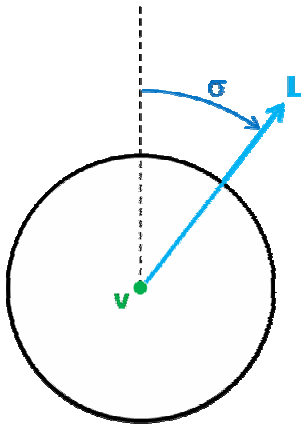


Figure 14: Bank angle definition.

The lift vector, **L**, is then a composition of two rotations—one rotation about the velocity vector by angle σ, followed by a 90° rotation about the angular momentum vector. The

thrust vector, **T**, is a vector in an arbitrary direction prescribed by the control law, and is of the form

$$\mathbf{T} = T_x \mathbf{i}^{\mathbf{I}} + T_y \mathbf{j}^{\mathbf{I}} + T_z \mathbf{k}^{\mathbf{I}} \tag{13}$$

where:

Tx. Ty. Tz = magnitude of thrust force in each Cartesian direction
$\mathbf{i}^{\mathbf{I}}$, $\mathbf{j}^{\mathbf{I}}$, $\mathbf{k}^{\mathbf{I}}$ = unit vectors along each Cartesian direction expressed in the inertial frame

The drag vector, **D**, is given by

$$\mathbf{D} = -qC_D S \frac{\mathbf{v}^{\mathbf{Rel}}}{\left|\mathbf{v}^{\mathbf{Rel}}\right|} \tag{14}$$

where:

C_D = coefficient of drag, discussed in the System Aerodynamics section
$\mathbf{v}^{\mathbf{rel}}$ = velocity vector of the vehicle relative to the atmosphere

The time variation of the last state, mass, is given by

$$\dot{m} = -\frac{\left|\mathbf{T}\right|}{g_0 I_{SP}} \tag{15}$$

where:

g_0 = 9.806 m/s^2, a constant

The form of the equations of motion used in PESST assumes that atmosphere rotates with the planet, therefore the speed of the atmosphere varies linearly with altitude. Additionally, a spherical planet with no wind is utilized in the simulation of the trajectory, although the planet does rotate at a constant angular rate.

5. Thermal Response

The TPS Sizer uses the Sutton-Graves convective heating relation for convective heating and the Tauber-Sutton correlation for Earth and Mars to determine radiative heating.

$$\dot{q}_{conv} = k\left(\frac{\rho}{r_{nose}}\right)^{1/2} v^3 \tag{16}$$

$$\dot{q}_{rad} = C\, r_{nose}^{a}\, \rho_{\infty}^{b}\, f(v) \tag{17}$$

where:

k = 1.74153x10^{-4} (for Earth) or 1.9027x10^{-4} (for Mars)
$f(v)$ is a tabulated function of velocity

There is a steady state ablation assumption made that uses the heat of ablation to estimate the recession during entry. There is also a finite-difference calculation to determine a one dimensional in-depth temperature response to the experienced heating. The temperature response is used to estimate the amount of TPS material required as insulation to keep the bondline temperature below its passed in limit.

6. Weights and Sizing

The weights and sizing module of PESST uses mass estimating relationships (MERs) to obtain the masses for the major components of the entry system—the thermal protection system (TPS) and its structure, the backshell, the propulsion system, and the aerodynamic deceleration system. These system masses are sized using fixed point iteration with relaxation until the desired landed mass is achieved. As shown in the DSM for PESST, Figure 1, the thermal response is dependent on the trajectory flown; therefore, for each iteration of the sizer, a new trajectory (starting from the same initial conditions) is flown to obtain the appropriate thermal response of the system to size the heatshield.

a) Thermal Protection System Sizing

The TPS is assumed to be comprised of three main components—the heatshield, the insulation, and the substructure required to support the heatshield and insulation. The thickness of the heatshield is assumed to be uniform across the vehicle's forebody and is sized to the stagnation point heating requirements determined by the thermal response analysis described previously. Three heatshield materials are available for selection—SLA-561, PICA, and carbon-carbon. The density for each of the three materials is shown in Table 1.

Table 1: Heat material densities.

Material	Density Used (kg/m3)
SLA-561	264.3
PICA	227.0
Carbon-Carbon	1890.0

The thickness of the insulation is uniformly sized to the thickness required to maintain the bondline temperature specified by the user in the ***Thermal*** tab of the GUI. A single material, Fiber-Form©, is available for the insulation material. Fiber-Form© has a density, $\rho_{Insulation}$, of 180 kg/m^3. The substructure associated with the heatshield is sized according to a constant mass fraction given by the relation

$$m_{Structure} = 0.08 m_{Entry} \tag{18}$$

where m_{Entry} is the entry mass at a given iteration. The total TPS mass at a given iteration can then be computed as

$$m_{TPS} = m_{Heatshield} + m_{Insulation} + m_{Structure} = At_{Heatshield}\rho_{Heatshield} + At_{Insulation}\rho_{Insulation} + 0.08 m_{Entry} \tag{19}$$

where:

A = forebody surface area
$t_{Heatshield}$ = heatshield thickness
$t_{Insulation}$ = insulation thickness

b) Propulsion System Sizing

Given the type of fuel, the type of propellant tank material, maximum thrust required and the engine type the mass of the propulsion system is determined.

Table 2: Fuel densities.

Fuel Used	Density Used (kg/m3)
MMH	878.0
NTO	1440.0
LOX	1142.0
LH2	71.0
RP1	810.0
CH4	422.6

Engines with pressurization systems will also have the required helium gas and tank sized.

$$mass_{propulsion\ \ system} = mass_{pressurization\ system} + mass_{\tan ks} + mass_{engine} + mass_{lines\ and\ valves} \tag{20}$$

c) Parachute System Sizing

This sizing module estimates the mass of the parachute and mortar for a Disk Gap Band parachute.

These relationships are valid in sizing DGB parachutes and a great amount of material can be found in *Parachute Recovery Systems* by T.W. Knacke.

$$number\ of\ \ gores = diameter \cdot 2 + 4 \tag{21}$$

$$line\ length = 1.15 \cdot diameter \tag{22}$$

The maximum force that will be seen by the cluster of parachutes is determined with a modified C_D which is determined from parachute clustering.

$$C_D = \left(1.05 - 0.05 \cdot number\ of\ \ parachutes\right) \cdot C_{D\ \ one\ parachute} \tag{23}$$

$$Force_{\max} = C_K \cdot dynamic\ \ pressure \cdot C_D \cdot surface\ area \tag{24}$$

$$Force_{designed\ load} = 1.5 \cdot Force_{\max} \tag{25}$$

The mass of the parachute is determined as:

$$mass_{parachute} = mass_{radials} + mass_{lines} + mass_{riser} + mass_{fabric} \quad (26)$$

A single mortar is sized for this parachute that weighs 1.48 times the weight of the parachute it releases.

E. Example Cases for PESST

1. Mars Pathfinder

Mars Pathfinder was a mission launched in 1996 with its 70° sphere-cone aeroshell carrying a 360 kg payload consisting of a lander and the Sojourner rover arriving at Mars on July 4, 1997. For validation purposes, PESST was used to size the entry system for the Pathfinder system. The inputs for each tab of the PESST GUI are shown in Table 7 and Table 8. It can be seen that the system is modeled with a 12.5 m parachute deploying at a dynamic pressure of 585 Pa, followed by heatshield jettison at a dynamic pressure of 31.98 kg. The parachute use is discontinued at -2410 m and the backshell is jettisoned at -2412 m. The actual flight system for Pathfinder included solid retrorockets to negate the residual velocity before the bridle was cut; however PESST does not contain any solid propellant models, so this is excluded within the PESST comparison.

Table 3: PESST inputs for Pathfinder comparison.

	Inputs	*Value*
Atmosphere		
	Attracting Body	Mars
	Atmospheric Models	Pathfinder
	Surface Gravity [m/s2]	3.718
	Surface Gas Density [kg/m2]	0.0068
	Scale Height [m]	17391.3
	Planet Radius [m]	3397200
	Planet Mass [kg]	6.42E+23
	Rotation Rate [deg/s]	0.004061239
	Perfect Gas, Specific Heat Ratio	1.3
Entry Body		
	Entry Mass [kg]	584
	Landed Mass Desired [kg]	360
	Entry Body Type	Sphere Cone
	Nose Radius [m]	0.6638
	Forebody Section One, Cone Angle [deg]	70
	Forebody Section One, Maximum Diameter [m]	2.7
	Area Convered by TPS [m2]	5.8640707
Trajectory		
	Entry Altitude [m]	128000
	Entry Velocity [m/s]	7479
	Entry Flight Path Angle [deg]	-13.65
	Entry Azimuth [deg]	253.7
	Entry Latitude [deg N]	22.63
	Entry Longtidue [deg E]	338
	Entry Angle-of-Attack [deg]	0
	Entry Bank Angle [deg]	0

Table 4: PESST inputs for Pathfinder comparison.

	Inputs	*Value*
Thermal		
	Heatshiel Material	SLA-561
	Stagnation Point Radius [m]	0.6638
	Initial Temperature of TPS [ºC]	75
	Bondline Temperature Limit [ºC]	250
	Recession Threshold Temp [ºC]	1500
	Thickness Guess [cm]	1
	Maximum Grid Size [cm]	0.1
	Time Step [s]	0.5
	Bondline Temp Convergence Tolerance [ºC]	0.25
	Relaxation Factor	1
	Iteration Limit	25
Events		
	Instantaneous Mass Drop 1, Start Dynamic Pressure [Pa]	31.98
	Instantaneous Mass Drop 1, Mass	Heatshield
	Instantaneous Mass Drop 2, Start Altitude [m]	-2412
	Instantaneous Mass Drop 2, Mass	Backshell
	Aerodynamics Dominated by Parachute, Start Dynamic Pressure [Pa]	585
	Aerodynamics Dominated by Parachute, End Altitude [m]	-2410
	Aerodynamics Dominated by Parachute, Parachute Diameter [m]	12.5
	Aerodynamics Dominated by Parachute, Aerodynamics File	Viking DGB Parachute
Options		
	Maximum Trajectory Time [s]	2000
	Minimum Trajectory Altitude [m]	-2500
	Maximum Trajectory Altitude [m]	1300000
	Modified Newtonian Aerodynamics	Yes
	Converge Design	Yes
	Size Thermal Protection System	Yes

Table 5 shows a comparison of the Mars Pathfinder trajectory as modeled in POST and in PESST, respectively. It is seen that the two trajectories have reasonable correspondence, with even altitudes differing the most between the systems. This error is caused by slight inconsistencies resulting from the meshing of the entry vehicle, a less detailed drag coefficient curve for the parachute, and the parachute inflation being assumed to be instantaneous.

Table 5: Pathfinder trajectory comparison.

Event	*POST*	*PESST*	*% Difference*
Entry			
Time [s]	0	0	0.00
Altitude [m]	128000	128000	0.00
Relative Velocity [m/s]	7479	7479	0.00
Relative Flight Path Angle [deg]	-13.65	-13.65	0.00
Parachute Deploy			
Time [s]	154.5	149.2	-3.43
Altitude [m]	9916	11496	15.93
Relative Velocity [m/s]	414.5	444.12	7.15
Dynamic Pressure [Pa]	585.0	584	-0.17
Heatshield Jettison			
Time [s]	174.5	171	-2.01
Altitude [m]	8219	9633	17.20
Relative Velocity [m/s]	90.23	95.68	6.04
Dynamic Pressure [Pa]	31.98	31.92	-0.19
Trajectory Termination			
Time [s]	359.8	380	5.61
Altitude [m]	-2408	-2410.36	0.10
Relative Velocity [m/s]	42.64	42.52	-0.28
Dynamic Pressure [Pa]	21.55	20.61	-4.36

Table 11 shows a comparison of the mass of the actual Pathfinder flight system and that obtained by PESST. Excluding the terminal propulsion system causes roll-up effects to be seen at the entry mass level. Additionally, the mass sizing for PESST for the backshell includes additional structure needed in the aeroshell; whereas, the mass cited for the flight system does not include this additional structure. However, the other system masses are acceptably close for conceptual design purposes to the actual flight system.

Table 6: Pathfinder mass comparison.

Element	*Flight System*	*PESST*	*% Difference*
Entry Mass [kg]	585.3	527	9.96
Heatshield [kg]	64.4	71	-10.25
Backshell and Structure [kg]	56.9	89	-56.41
Parachute [kg]	9.75	7	28.21
Propulsion System [kg]	30.7	N/A*	N/A
Landed Mass [kg]	360	360	0.00

*Solid Rocket Propulsion System Not Sized in PESST

2. Mars Pinpoint Landing Comparison

To analyze the cost of pinpoint landing, two different guidance algorithms were analyzed on a high-mass robotic class mission to Mars. All of the parameters input into PESST were identical except for the guidance event. One event used a gravity turn guidance law targeting an altitude of 0 m and a velocity of 1 m/s, whereas the other used the analytic terminal descent guidance law as described previously, targeting 0 m, 1 m/s, and downrange 10 km from the gravity turn landed position. Effectively, the gravity turn case

targets a soft touchdown on the surface, whereas the analytic terminal descent guidance law targets a specific location. The inputs into PESST for each case are seen in Table 7 through Table 10 for the gravity turn and pinpoint case, respectively.

Table 7: PESST inputs for the gravity turn.

	Inputs	*Value*
Atmosphere		
	Attracting Body	Mars
	Atmospheric Models	Pathfinder
	Surface Gravity [m/s2]	3.718
	Surface Gas Density [kg/m2]	0.0068
	Scale Height [m]	17391.3
	Planet Radius [m]	3397200
	Planet Mass [kg]	6.42E+23
	Rotation Rate [deg/s]	0.004061239
	Perfect Gas, Specific Heat Ratio	1.3
Entry Body		
	Entry Mass [kg]	2920
	Landed Mass Desired [kg]	1590
	Entry Body Type	Sphere Cone
	Nose Radius [m]	1.25
	Forebody Section One, Cone Angle [deg]	70
	Forebody Section One, Maximum Diameter [m]	4.5
	Area Convered by TPS [m2]	16.906017
Trajectory		
	Entry Altitude [m]	128000
	Entry Velocity [m/s]	6000
	Entry Flight Path Angle [deg]	-15.2
	Entry Azimuth [deg]	90
	Entry Latitude [deg N]	0
	Entry Longtidue [deg E]	0
	Entry Angle-of-Attack [deg]	-15
	Entry Bank Angle [deg]	0

Table 8: PESST inputs for the gravity turn.

	Inputs	*Value*
Thermal		
	Heatshiel Material	PICA
	Stagnation Point Radius [m]	1.25
	Initial Temperature of TPS [°C]	75
	Bondline Temperature Limit [ºC]	250
	Recession Threshold Temp [ºC]	1500
	Thickness Guess [cm]	1
	Maximum Grid Size [cm]	0.1
	Time Step [s]	0.5
	Bondline Temp Convergence Tolerance [ºC]	0.25
	Relaxation Factor	1
	Iteration Limit	25
Events		
	Instantaneous Mass Drop 1, Start Altitude [m]	4000
	Instantaneous Mass Drop 1, Mass	Heatshield
	Instantaneous Mass Drop 2, Start Altitude [m]	3500
	Instantaneous Mass Drop 2, Mass	Backshell
	Aerodynamics Dominated by Parachute, Start Mach [-]	3
	Aerodynamics Dominated by Parachute, End Altitude [m]	3500
	Aerodynamics Dominated by Parachute, Parachute Diameter [m]	19.7
	Aerodynamics Dominated by Parachute, Aerodynamics File	Viking DGB Parachute
	Guidance, Start Altitude [m]	3000
	Guidance, End Altitude [m[	10
	Guidance, Type	Gravity Turn
	Guidance, Target Altitude [m]	0
	Guidance, Target Velocity [m/s]	1
	Guidace, Propellant Type	Presure-fed Hydrazine
	Guidance, Propellant Tank Material	Titanium
	Guidance, Specific Impulse of Engines [s]	194
	Guidance, Max Thrust from Engines [N]	25000
Options		
	Maximum Trajectory Time [s]	2000
	Minimum Trajectory Altitude [m]	0
	Maximum Trajectory Altitude [m]	1300000
	Modified Newtonian Aerodynamics	Yes
	Converge Design	Yes
	Size Thermal Protection System	Yes

Table 9: PESST inputs for the analytic guidance law case.

Inputs		*Value*
Atmosphere		
	Attracting Body	Mars
	Atmospheric Models	Pathfinder
	Surface Gravity [m/s2]	3.718
	Surface Gas Density [kg/m2]	0.0068
	Scale Height [m]	17391.3
	Planet Radius [m]	3397200
	Planet Mass [kg]	6.42E+23
	Rotation Rate [deg/s]	0.004061239
	Perfect Gas, Specific Heat Ratio	1.3
Entry Body		
	Entry Mass [kg]	2920
	Landed Mass Desired [kg]	1590
	Entry Body Type	Sphere Cone
	Nose Radius [m]	1.25
	Forebody Section One, Cone Angle [deg]	70
	Forebody Section One, Maximum Diameter [m]	4.5
	Area Convered by TPS [m2]	16.906017
Trajectory		
	Entry Altitude [m]	128000
	Entry Velocity [m/s]	6000
	Entry Flight Path Angle [deg]	-15.2
	Entry Azimuth [deg]	90
	Entry Latitude [deg N]	0
	Entry Longtidue [deg E]	0
	Entry Angle-of-Attack [deg]	-15
	Entry Bank Angle [deg]	0

Table 10: PESST inputs for the analytic guidance law case

	Inputs	*Value*
Thermal		
	Heatshiel Material	PICA
	Stagnation Point Radius [m]	1.25
	Initial Temperature of TPS [ºC]	75
	Bondline Temperature Limit [ºC]	250
	Recession Threshold Temp [ºC]	1500
	Thickness Guess [cm]	1
	Maximum Grid Size [cm]	0.1
	Time Step [s]	0.5
	Bondline Temp Convergence Tolerance [ºC]	0.25
	Relaxation Factor	1
	Iteration Limit	25
Events		
	Instantaneous Mass Drop 1, Start Altitude [m]	4000
	Instantaneous Mass Drop 1, Mass	Heatshield
	Instantaneous Mass Drop 2, Start Altitude [m]	3500
	Instantaneous Mass Drop 2, Mass	Backshell
	Aerodynamics Dominated by Parachute, Start Mach [-]	3
	Aerodynamics Dominated by Parachute, End Altitude [m]	3500
	Aerodynamics Dominated by Parachute, Parachute Diameter [m]	19.7
	Aerodynamics Dominated by Parachute, Aerodynamics File	Viking DGB Parachute
	Guidance, Start Altitude [m]	3000
	Guidance, End Altitude [m[	10
	Guidance, Type	D'Souza
	Guidance, Target Altitude [m]	0
	Guidance, Target Velocity [m/s]	1
	Guidance, Target Latitude [deg]	0
	Guidance, Target Longtiude [deg]	14.9277
	Guidance, Target Flight Path Angle [deg]	-90
	Guidance, Target Heading Angle [deg]	90
	Guidace, Propellant Type	Presure-fed Hydrazine
	Guidance, Propellant Tank Material	Titanium
	Guidance, Specific Impulse of Engines [s]	194
	Guidance, Max Thrust from Engines [N]	25000
Options		
	Maximum Trajectory Time [s]	2000
	Minimum Trajectory Altitude [m]	0
	Maximum Trajectory Altitude [m]	1300000
	Modified Newtonian Aerodynamics	Yes
	Converge Design	Yes
	Size Thermal Protection System	Yes

Table 11 shows a comparison of the subsystem masses for the two cases. It can be seen that the gravity turn case required a propellant mass fraction, defined as the ratio of propellant mass to entry mass, of 0.094 whereas the pinpoint landing case has a propellant mass fraction of 0.153 (a 61.7% increase). However, the net effect on the entry mass in 9.03%. Thus, for the specific case analyzed, the additional mass required to perform pinpoint landing as opposed to a soft surface landing is 242 kg (9.03%).

Table 11: Soft versus pinpoint landing subsystem mass comparison.

Element	*Gravity Turn*	*Pinpoint Landing*	*% Difference*
Entry Mass [kg]	2681	2923	9.03
Heatshield [kg]	390	398	2.05
Backshell and Structure [kg]	167	169	1.20
Parachute [kg]	34	34	0.00
Propulsion System [kg]	247	286	15.79
Propellant [kg]	253	446	76.28
Landed Mass [kg]	1590	1590	0.00

3. Stardust

The tool was exercised to size the vehicle of the Stardust mission. The parameters associated with this mission may be seen in Table 12 and Table 13. Only the inertial velocity magnitude and inertial entry flight path angle was obtained. In order to obtain reasonable planet relative entry values required by the tool, a planar equatorial entry was assumed. This is different than the actual entry that terminated at the Utah Test and Training Range but provides a reasonable entry profile required for vehicle sizing. The payload mass of Stardust was not found and is likely due to the vehicle remaining intact (without heatshield jettison or main parachute jettison) through ground impact. An assumed payload mass of 5 kg was used to estimate the Aerogel, container, and collected debris. PESST was used to size Stardust and resulted in an entry mass of 43.12 kg as compared to the actual entry mass of 45.8 kg.

Table 12: Stardust Mission Inputs

Inputs		*Value*
Atmosphere		
	Attracting Body	Earth
	Atmospheric Models	US 1976
	Surface Gravity [m/s2]	9.81
	Surface Gas Density [kg/m2]	1.2260066
	Scale Height [m]	7257
	Planet Radius [m]	6378136
	Planet Mass [kg]	5.97E+24
	Rotation Rate [deg/s]	0.004178008
	Perfect Gas, Specific Heat Ratio	1.4
Entry Body		
	Entry Mass [kg]	45.8
	Landed Mass Desired [kg]	5
	Entry Body Type	Sphere Cone
	Nose Radius [m]	0.2202
	Forebody Section One, Cone Angle [deg]	59.5
	Forebody Section One, Maximum Diameter [m]	0.8
	Area Convered by TPS [m2]	0.596
Trajectory		
	Entry Altitude [m]	125000
	Entry Velocity [m/s]	12.4439
	Entry Flight Path Angle [deg]	-8.5027
	Entry Azimuth [deg]	90
	Entry Latitude [deg N]	0
	Entry Longtidue [deg E]	0
	Entry Angle-of-Attack [deg]	0
	Entry Bank Angle [deg]	0

Table 13: Stardust Mission Inputs

	Inputs	*Value*
Thermal		
	Heatshiel Material	PICA
	Stagnation Point Radius [m]	0.2202
	Initial Temperature of TPS [ºC]	-75
	Bondline Temperature Limit [ºC]	250
	Recession Threshold Temp [ºC]	1500
	Thickness Guess [cm]	1
	Maximum Grid Size [cm]	0.1
	Time Step [s]	0.5
	Bondline Temp Convergence Tolerance [ºC]	0.25
	Relaxation Factor	1
	Iteration Limit	25
Events		
	Aerodynamics Dominated by Parachute, Mach	1.37
	Instantaneous Mass Drop 1, Altitude [m]	1
Options	Maximum Trajectory Time [s]	1000
	Minimum Trajectory Altitude [m]	0
	Maximum Trajectory Altitude [m]	1300000
	Modified Newtonian Aerodynamics	Yes
	Converge Design	Yes
	Size Thermal Protection System	Yes

F. Creating a PESST Development Environment for Windows

1. Helpful tools

0) If in need of an open-sourced zip program; 7-Zip is an excellent option.
0a) Textpad is an excellent editor able to read files with both Unix and MS return characters easily

2. Installation Instructions for a PESST developer

1) Download and install the latest available Java Development Kit (JDK)
http://java.sun.com/javase/downloads/index.jsp Get 'JDK 6 Update 6'
1a) Open a **new** command prompt window and type 'java -version'
java version "1.6.0_06"
Java(TM) SE Runtime Environment (build 1.6.0_06-b02)
Java HotSpot(TM) Client VM (build 1.6.0_06-b02, mixed mode, sharing)

2) Download and install gfortran compiler for the **Native Windows** environment.
http://gcc.gnu.org/wiki/GFortranBinaries

Test for correct installation
2a) Open a **new** command prompt and type 'gfortran --version'
You should see something like this:
GNU Fortran (GCC) 4.4.0 20080415 (experimental) [trunk revision 134312]

Copyright (C) 2008 Free Software Foundation, Inc.

GNU Fortran comes with NO WARRANTY, to the extent permitted by law.
You may redistribute copies of GNU Fortran
under the terms of the GNU General Public License.
For more information about these matters, see the file named COPYING

2b) Type 'gcc --version'
You should see something like this:
gcc (GCC) 4.4.0 20080415 (experimental) [trunk revision 134312]
Copyright (C) 2008 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

3) Download the Apache Ant resource framework. Ant can be thought of as a very full featured replacement for 'make'.
http://ant.apache.org/
Unzip binary into user specified folder; this guide will ***assume*** the ant folder was placed under **C:\utilities**

3a) Install Ant

http://ant.apache.org/manual/install.html#installing

Refer to the link as an installation resource but it boils down to:
'add to' or create the following environmental variables:
ANT_HOME=C:\utilities\apache-ant-1.7.0 <match to your location>
JAVA_HOME=C:\Program Files\Java\jdk1.<version> <match to your location>
PATH= <Your current path entries>;%ANT_HOME%\bin

There are a few of ways to go about setting these environment variables. You can either do it directly from the command prompt, or via the standard Windows method using a Windows GUI interface.

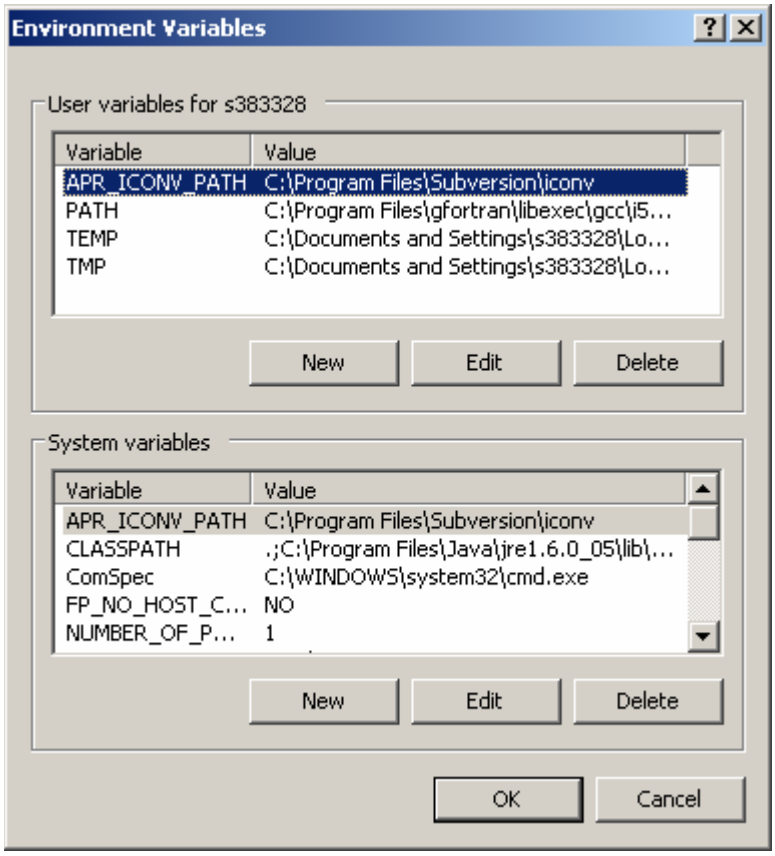
Following the setting of these environment variables, a restart will most likely be required.

First, for the command prompt method…

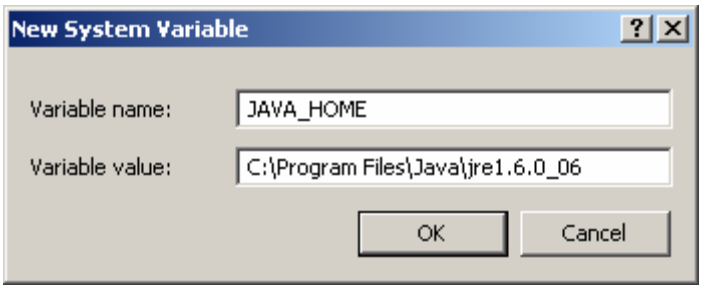
1) Open a new command prompt by your preferred method.
2) Assume Ant is installed in c:\utilities\ as above. The following sets up the environment
3) set ANT_HOME=c:\utilities
4) set JAVA_HOME=c:\Program Files\Java\jdk1.<version> <match to your location>
5) set PATH=%PATH%;%ANT_HOME%\bin

Now, for the Windows GUI method…

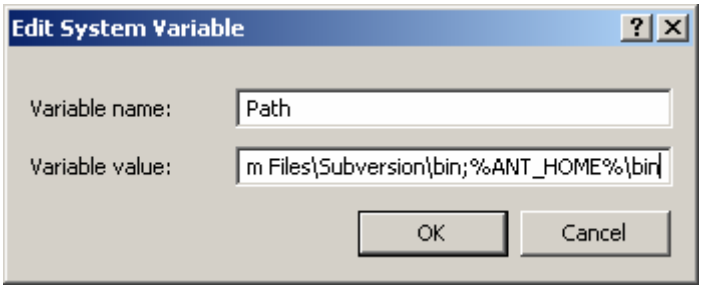
1) Right click on your "My Computer" icon on your desktop and select properties
2) Select the "Advanced" tab and click the "Environment Variables" button at the bottom of the dialog box. This should bring up the Environment Variables dialog box below.



3) The top portion of this dialog box are the environment variables that are specific to your user account and the bottom is for all users. You should be able to update the required environment variables in either set without any issues. For the purpose of this walk-through, use the "System Variables" frame on the bottom to set up your environment variables.
4) If you haven't already added the environment variables via the command line method provided above, click the "New" button to add the "JAVA_HOME" and "ANT_HOME" variables with values as before.



5) After you have added these variables, scroll through the list to find the "PATH" environment variable. Select it and click the edit button. In the "Edit System Variable" dialog box, scroll to the end of the "Variable value:" field and append ";%ANT_HOME%\bin".



6) After you have completed this step, save these settings by selecting the OK button and closing the System Properties dialog box and then it will probably be best to restart your system to ensure the environment variables are updated.

3b) To test installation, open a ***NEW*** command prompt:

> **echo %JAVA_HOME%**
C:\Program Files\Java\jdk1.6.0_03

> **echo %ANT_HOME%**
C:\utilities\apache-ant-1.7.0

> **echo %PATH%**
<whatever is in your path>;C:\utilities\apache-ant-1.7.0\bin

> **ant -version**
Apache Ant version 1.7.0 compiled on December 13 2006

4) Download and install Netbeans to modify the GUI or Java code. The development environment has excellent GUI design capabilities.

http://www.netbeans.org/

5) To build the Windows executable

a) Open the nbproject/project.properties file and make certain that the path to the matlab home and gfortran home directories is correct for your system.

b) Open a command prompt window

c) Go to the root pesst folder and run the command 'ant'

d) Read through the output from ant to make sure that there were no errors. Build successful only means that the java code for the GUI was successfully compiled.

e) change to the 'dist' directory. This directory has the compiled executables.

f) To start the GUI type 'java –jar pesst.jar'

or

f) To see the inputs required by either the generateAero or the runPesst executables just run them without arguments to get help info on running them from the command line.

Question: **When I try building PESST by using ant I see this message at the top of a stream of errors**.

-pre-jar:
[exec] native\wrappers\aeromesh.c:1:18: error: math.h: No such file or directory
[exec] native\wrappers\aeromesh.c:2:20: error: locale.h: No such file or directory
[exec] native\wrappers\aeromesh.c:3:19: error: stdio.h: No such file or directory
[exec] native\wrappers\aeromesh.c:4:20: error: stdlib.h: No such file or directory
[exec] native\wrappers\aeromesh.c:5:20: error: string.h: No such file or directory

Answer: You'll have to open the project.properties file under the nbproject folder. At the top, change the path to your gfortran/matlab file to one that matches your system.

G. Acknowledgements

The PESST framework has been under development within the Space Systems Design Laboratory (SSDL) at Georgia Tech for the past three years. Support for the PESST framework has been provided by NASA Langley Research Center, the Jet Propulsion Laboratory and the Charles Stark Draper Laboratory. The authors extend their gratitude to these three institutions for their insight and feedback throughout the PESST development cycle. Additional thanks are extended to the following SSDL graduate students for their contributions to PESST: Devin Kipp (JPL), John Dec (NASA Langley), Grant Wells and John Theisinger.

II. Guidance, Navigation, and Control System Trades for Mars Pinpoint Landing

Guidance, Navigation, and Control Technology System Trades for Mars Pinpoint Landing

Bradley A. Steinfeldt[1]
Georgia Institute of Technology, Atlanta, GA, 30332

Landing site selection is a compromise between safety concerns associated with the site's terrain and scientific interest. Therefore, technologies enabling pinpoint landing (sub-100 m accuracies) on the surface of Mars are of interest to increase the number of accessible sites for *in-situ* research as well as allow placement of vehicles nearby prepositioned assets. A survey of various guidance, navigation, and control technologies that could allow pinpoint landing to occur at Mars has shown that negligible propellant mass fraction benefits are seen for reducing the three-sigma position dispersion at parachute deployment below approximately 3 km. Four different propulsive terminal descent guidance algorithms were analyzed with varying applicability to flight. Of these four, a near propellant optimal, analytic guidance law showed promise for the conceptual design of pinpoint landing vehicles. In addition, subsonic guided parachutes are shown to provide marginal performance benefits due to the timeline associated with Martian entries, and a low computational-cost, yet near fuel optimal propulsive terminal descent algorithm is identified. This investigation also demonstrates that navigation is a limiting technology for Mars pinpoint landing, with landed performance being largely affected by sensor accuracy.

Nomenclature

a_i = Acceleration along the i^{th} direction
a_j = Scalar defining convex state constraints
$\mathbf{a}$ = Acceleration vector $= \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix}^T$
b = Scalar weighting parameter
C_{j_i} = j^{th} constant coefficient used in the modified Apollo lunar module guidance algorithm
dt_f = Terminal time increment
$\mathbf{f}$ = Set of first-order differential equations of motion
g = Local acceleration due to gravity
$\mathbf{g}$ = Acceleration vector due to gravity
i = Index
I_{JJ} = Partition used in the optimal-control solution
$I_{J\psi}$ = Partition used in the optimal-control solution
$I_{\psi J}$ = Partition used in the optimal-control solution
$I_{\psi\psi}$ = Partition used in the optimal-control solution
J = Performance index
L = Scalar objective function describing path parameters
$\mathbf{p}$ = Influence function vector

[1] Graduate Research Assistant, Guggenheim School of Aerospace Engineering, 270 Ferst Drive, Atlanta, GA 30332

r_i = Position along the i^{th} direction
$\mathbf{r}$ = Position vector $= \begin{pmatrix} r_1 & r_2 & r_3 \end{pmatrix}^T$
R = Matrix of influence functions
S_j = Matrix defining convex state constraints
t = Time
t_{go} = Time-to-go until touchdown
$\mathbf{u}$ = Control vector
$\delta\mathbf{u}$ = Control vector increment
v_i = Velocity along the i^{th} direction
$\mathbf{v}$ = Velocity vector $= \begin{pmatrix} v_1 & v_2 & v_3 \end{pmatrix}^T$
W = Positive-definite weighting matrix
$\mathbf{x}$ = State vector $= \begin{pmatrix} \mathbf{r}^T & \mathbf{v}^T & m \end{pmatrix}^T$

α = Mass consumption rate
Γ = Weighting on final time to go
ε = Tolerance level
ζ = Slack variable bounding thrust magnitude
ρ_1 = Thrust magnitude lower bound
ρ_2 = Thrust magnitude upper bound
$\boldsymbol{\tau}_c$ = Commanded thrust vector
υ_j = Vector defining convex state constraints
ϕ = Scalar objective function
$\boldsymbol{\psi}$ = Adjoint constraint equations

CDF = Cumulative Distribution Function
EDL = Entry, Descent, and Landing
EI = Entry Interface
GNC = Guidance, Navigation, and Control
MER = Mars Exploration Rovers
MSL = Mars Science Laboratory
PMF = Propellant Mass Fraction
SOCP = Second-order Cone Problem
TCM = Trajectory Correction Maneuver
TRN = Terrain Relative Navigation

A. Introduction

At present, the choice of landing sites for Mars exploration vehicles is a trade between scientific interest and landing safety in which the safety element may preclude many interesting regions of the planet. The landed accuracy of an entry system is a function of four major items—delivery error at entry interface (EI), knowledge uncertainty at EI, environmental uncertainty, and vehicle performance[1]. Delivery error at EI refers to how closely the vehicle's actual position and velocity vector at EI match the desired EI position and velocity vectors and is driven primarily by interplanetary navigation and how accurately trajectory correction maneuvers (TCMs) are performed. Knowledge uncertainty at EI is a result of accumulated sensor error from the last navigational update as well as the accuracy of that navigation update. Environmental uncertainty consists primarily of atmospheric deviations from the nominal density and wind profiles through the atmosphere, although other sources such as gravitational field modeling impact this uncertainty as well. The dispersions associated with the performance of the vehicle are comprised of uncertainties in the physical model of the entry system—mass properties, aerodynamic characteristics, etc., and how its systems perform—deployment events associated with the parachute, performance of the guidance, navigation, and control (GNC) systems, thrust and duration of burns, etc. The landing ellipse, as shown in Figure 1, is the cumulative effect of these uncertainties propagated throughout the vehicle's entire trajectory mapped to a physical location on the surface of the destination planet. The Mars Science

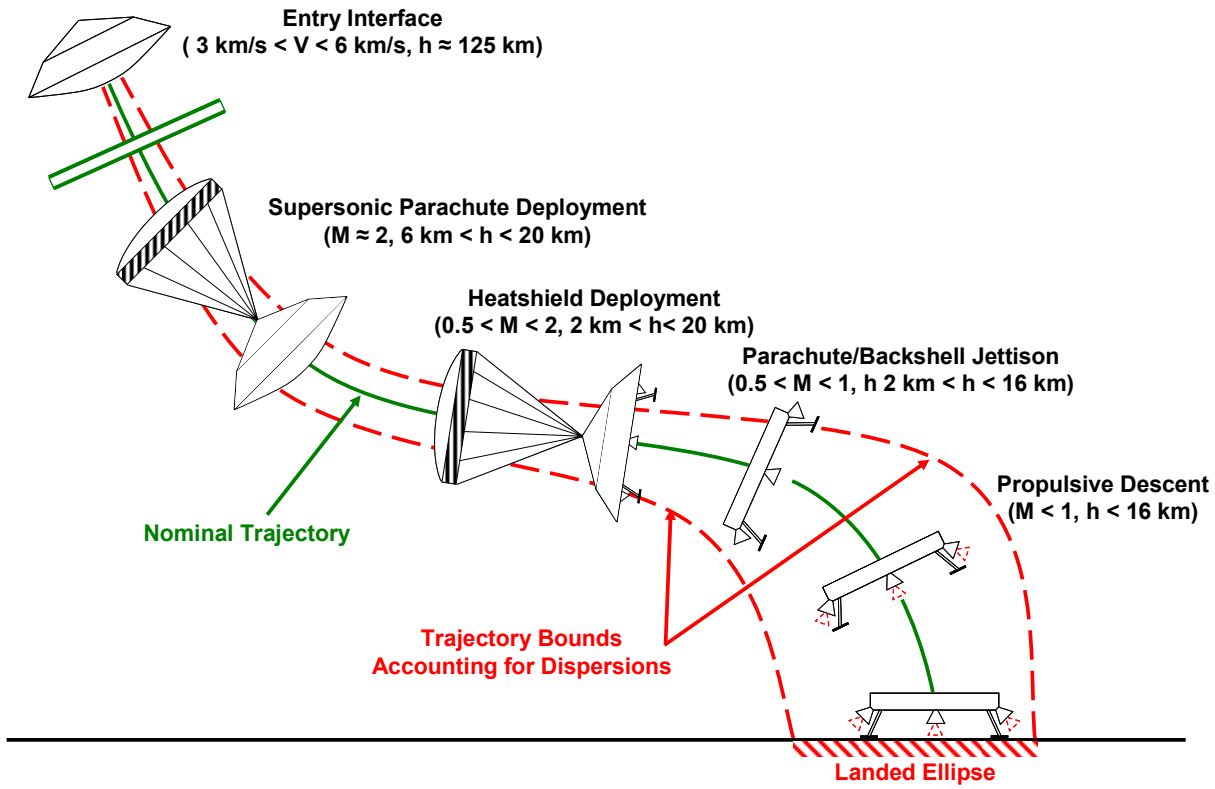


Figure 15. Typical Martian EDL sequence with uncertainty.

Laboratory (MSL), planned to launch in 2009, is anticipating a landing ellipse major axis of approximately 20 km which is a four-time reduction from the Mars Exploration Rovers (MER) and over an order of magnitude improvement from the Mars Pathfinder mission[2]. Relative to MER, this landed ellipse accuracy improvement is largely the result of the inclusion of a modified Apollo hypersonic guidance algorithm which modulates the direction of the vehicle's lift vector to accommodate uncertainties in the atmospheric flight path[3]. Pinpoint landing accuracy is defined as a further two-order of magnitude reduction to MSL's landing ellipse major axis to sub-100 m levels. By achieving this level of accuracy, a number of benefits can be realized such as minimizing rover traverse times to scientifically-rich locations and enabling entry systems to land near prepositioned assets on the surface as was outlined in the Mars Design Reference Mission for human exploration or suggested for robotic sample return missions[4].

B. Simulation and Vehicle Parameters

Various GNC technologies are studied on a large-scale robotic entry vehicle in order to understand their implications on the capability to achieve sub-100 m level landed accuracies. The technologies investigated span the entire EDL sequence from the hypersonic phase through terminal descent and touchdown. Hypersonic, subsonic parachute, and propulsive terminal descent guidance is investigated using ideal navigation and identifying the propellant mass fraction (PMF) required to achieve pinpoint level accuracy. Additionally, terrain relative navigation (TRN) is investigated by examining the effect of termination altitude and sampling frequency on TRN sensor performance as well as examining the effect of map-tie error on the overall landed vehicle's accuracy.

A three degree-of-freedom simulator with bank modulation is used for trajectory propagation. The simulator incorporates modular capability allowing easy incorporation of the various guidance algorithms assessed. The nominal trajectory has the vehicle, a 4.5 m, Viking heritage, 70° sphere-cone, starting at parachute deployment at an altitude of 8 km MOLA and Mach 2 with a flight path angle of -20°. It is assumed for all but the hypersonic guidance study, that the vehicle's initial state at parachute deployment has a state dispersion similar to that of MSL, implying that a modified Apollo guidance algorithm is used throughout the hypersonic phase of flight, except constant a constant parachute deployment altitude is assumed[5]. MarsGRAM was interrogated at a single latitude and longitude corresponding to the nominal

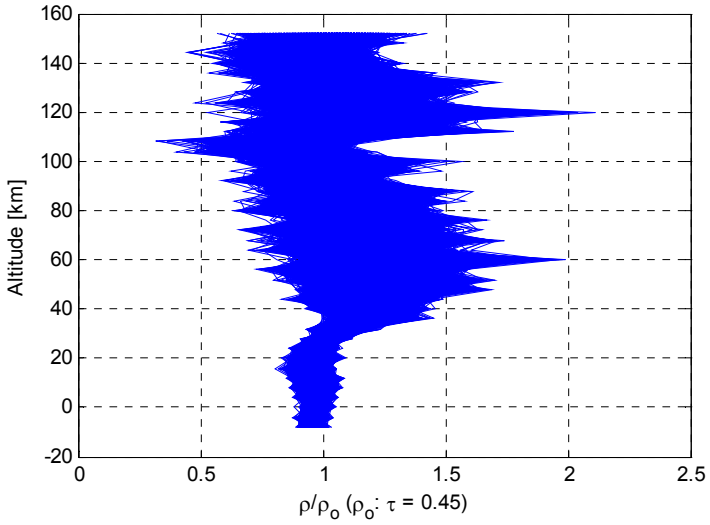


Figure 16. Density variation used in simulations.

Table 14. Vehicle and state parameters[1,5].

Parameter	*Nominal*	*Distribution*	*Deviation (3σ or min/max)*
Entry Mass	2616 kg	Gaussian	±3 kg
Entry Flight Path Angle	-14°	Gaussian	±0.6°
Vehicle Diameter	4.5 m	--	--
Trim Angle of Attack	11°	Gaussian	±2°
Parachute Deploy Distance from Nominal	0 km	Uniform	8 km
Parachute Deploy Velocity	488 m/s	Gaussian	1.3 m/s
Parachute Deploy Flight Path Angle	-20°	Uniform	±0.2°
C_A Multiplier (Kn≥0.1)	1	Gaussian	5%
C_N Multiplier (Kn≥0.1)	1	Gaussian	10%
C_A Multiplier (M>10)	1	Gaussian	3%
C_N Multiplier (M>10)	1	Gaussian	5%
C_A Multiplier (0.8<M<5)	1	Gaussian	10%
C_N Multiplier (0.8<M<5)	1	Gaussian	8%
C_A Multiplier (M<0.8)	1	Gaussian	5%
Supersonic Parachute Diameter	19 m	--	--
Supersonic Parchute C_D	0.61	Uniform	±10%
Subsonic Parachute Diameter	19 m	--	--
Subsonic Parchute C_D	0.68	Uniform	±10%
Maximum Terminal Descent Engine Thrust	3047 N	Uniform	±5%
Minimum Terminal Descent Engine Thrust*	1142 N	Uniform	±5%
Terminal Descent Engine I_{SP}	220 s	Uniform	±0.67%

landing site with dust tau varying between 0.1 and 0.9 to provide the mean and variation for the various environmental parameters used throughout the trajectory including the wind, acceleration due to gravity, and density[6]. Figure 2 shows a plot of the nominal density variation used in the simulations. Nominal vehicle, state, and environmental parameters with their dispersions are shown in Table 1. The entry state and hypersonic parameters were used by Striepe, *et al*, to derive the parachute deployment dispersion used for the initial conditions for the principal trades conducted[5].

C. Propulsive Terminal Descent

Four different propulsive terminal descent algorithms were evaluated in this study. The first of which is a modified Apollo lunar module terminal descent algorithm which assumes linear variation of the vertical

acceleration with quadratic variation in the remaining two axes and has no optimality conditions[7]. The second algorithm considered is a constrained gradient-based, indirect optimal control algorithm with iteration required to derive the control history[8]. The third algorithm, originally derived by D'Souza, is a fuel-optimal algorithm which assumes flight over a flat planet neglecting aerodynamic forces. These assumptions allow an analytic solution to be found which D'Souza showed to be optimal[9]. The fourth algorithm examined is a second-order cone formulation where convexity ensures that a global optimum is reached in a finite number of iterations with a feasible result obtained at each iteration, which is desirable should the algorithm be implemented on-board the vehicle[10].

1. Modified Apollo Lunar Module Terminal Descent Guidance Algorithm

The modified Apollo lunar module guidance algorithm begins by assuming that the acceleration profile is quadratic in each of the three directions (downrange, crossrange, and altitude) relative to the target[7]. In equation form, that is to say that the acceleration in each direction is given by

$$a_i(t) = C_{0_i} + C_{1_i} t + C_{2_i} t^2 \tag{1}$$

This can be integrated to give the velocity and distance variation with time in each axis

$$v_i(t) = C_{0_i} t + \frac{C_{1_i}}{2} t^2 + \frac{C_{2_i}}{3} t^3 + v_{0_i} \tag{2}$$

$$r_i(t) = \frac{C_{0_i}}{2} t^2 + \frac{C_{1_i}}{6} t^3 + \frac{C_{2_i}}{12} t^4 + v_{0_i} t + r_{0_i} \tag{3}$$

Evaluating Eqs. (1) – (3) at the initial conditions,

$$\mathbf{r}(t=0) = \mathbf{r_0} \text{ and } \mathbf{v}(t=0) = \mathbf{v_0} \tag{4 a-b}$$

and final conditions

$$\mathbf{r}(t=t_f) = \mathbf{r_f},\ \mathbf{v}(t=t_f) = \mathbf{v_f}, \text{ and } \mathbf{a}(t=t_f) = \mathbf{a_f} \tag{5 a-c}$$

allows the solution for the coefficients in each axis to be solved. The resulting coefficients are given by

$$C_{0_i} = a_{f_i} - \frac{6}{t_{go}}\left(v_{f_i} + v_{0i}\right) + \frac{12}{t_{go}^2}\left(r_{fi} - r_{0_i}\right) \tag{6}$$

$$C_{1_i} = -\frac{6}{t_{go}} a_{f_i} + \frac{6}{t_{go}^2}\left(5v_{f_i} + 3v_{0i}\right) - \frac{48}{t_{go}^3}\left(r_{fi} - r_{0i}\right) \tag{7}$$

$$C_{2_i} = \frac{6}{t_{go}^2} a_{f_i} - \frac{12}{t_{go}^3}\left(2v_{f_i} + v_{0_i}\right) + \frac{36}{t_{go}^4}\left(r_{f_i} - r_{0_i}\right) \tag{8}$$

By assuming a linear acceleration profile in the vertical axis (*i.e.*, setting C_2=0) the time-to-go, t_{go}, to be solved for analytically and is given by the expression

$$t_{go} = \frac{2v_{f_3} + v_{0_3}}{a_{f_3}} + \left[\left(\frac{2v_{f_3} + v_{0_3}}{a_{f_3}} \right)^2 + \frac{6}{a_{f_3}} \left(r_{0_3} - r_{f_3} \right) \right]^{1/2}, \; a_{f_3} \neq 0 \tag{9}$$

or

$$t_{go} = 3 \frac{r_{f_3} - r_{0_3}}{2v_{f_3} + v_{0_3}}, \; a_{f_3} = 0 \tag{10}$$

Thus, the commanded thrust vector is given by

$$\boldsymbol{\tau}_{\mathbf{C}} = m\left(\mathbf{a} - \mathbf{g}\right) \tag{11}$$

The primary advantage of this algorithm is that it is computationally non-complex and allows for the acceleration profile to be found for all time. However, the algorithm does not provide for conditions to obtain the fuel optimal solution or constraints on the maximum commanded thrust. For some trajectories, these limitations can result in a very large relative PMF when the loop is closed around the guidance algorithm as a low altitude hover ensures that pinpoint accuracy is achieved.

2. Gradient Based Optimal Control Algorithm

The general optimal control problem is the process of finding the control history, $\mathbf{u}(t)$, and final time, t_f, that minimizes the performance index

$$J = \phi\left(\mathbf{x}(t_f), t_f\right) + \int_{t_0}^{t_f} L\left(\mathbf{x}(t), \mathbf{u}(t), t\right) dt \tag{12}$$

for a set of given set of system equations

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, \mathbf{u}, t\right) \tag{13}$$

that describe the physical system. For the terminal descent problem, the state variables of interest, namely the position and velocity vectors, are known at an unknown terminal time. The main difficulty associated with this type of problem is the free terminal time which increases the dimensions of the optimization problem to be solved. Often, the terminal time is thought of as an additional control parameter. Classical optimal control theory presents several solution methods for the class of problem with the terminal conditions being specified at a free terminal time including neighboring extremal methods, gradient methods, and quasi-linearization methods[8]. All three methods are iterative and rely on an initial solution that is modified through successive linearization. A gradient based approach allows for less stringent conditions to be imposed on the initial solution than other classical methods making it preferable for conceptual design for propulsive terminal descent. However, near the optimum, the number of iterations increases dramatically. The constraints associated with the terminal descent problem, namely the surface constraint and the maximum available thrust, can either be handled through penalty methods that penalize deviations from the constraints or by adjoining them to the objective function, with the later being implemented in this analysis. For the propulsive terminal descent problem, the states, $\mathbf{x}(t)$, are the position and velocity of the vehicle relative to the target and the control vector, $\mathbf{u}(t)$, is the magnitude and direction of the thrust, or equivalently, the acceleration of the vehicle. A maximum thrust magnitude and an altitude restriction to prevent subterranean trajectories provide the constraints for the problem. With no weighting on the final time, a quadratic performance index can be formulated in the form of Eq. (12), which is comprised of solely the integrated control vector, $\mathbf{u}(t)$

$$J = \frac{1}{2}\int_{t_0}^{t_f} \mathbf{u}(t)^T \mathbf{u}(t) dt \tag{14}$$

The solution algorithm for the gradient based method implemented for this study is as follows[8]:

1. Obtain the equations describing the motion of the vehicle, $\mathbf{f}(\mathbf{x},\mathbf{u},t)$
2. Determine the constraints for the problem, thrust magnitude and radius of the planet's surface, and form the adjoint constraint equations, $\boldsymbol{\psi}[\mathbf{x}(t),t]$
3. Estimate the control history, $\mathbf{u}(t)$, for the thrust vector and the terminal time, t_f
4. Integrate the equations of motion, Eq. (13) forward using the initial conditions, $\mathbf{x}(t_0)$, and estimated control history from Step 3 from t_0 to t_f. Record $\mathbf{x}$(t), $\mathbf{u}$(t), $\boldsymbol{\psi}[\mathbf{x}(t_f),t_f]$, $\left[\frac{d\phi}{dt} + L\right]_{t=t_f}$, and $\left[\frac{d\boldsymbol{\psi}}{dt}\right]_{t=t_f}$
5. Integrate backwards in time the equations

$$\dot{\mathbf{p}} = -\left(\frac{d\mathbf{f}}{d\mathbf{x}}\right)^T \mathbf{p} - \left(\frac{dL}{d\mathbf{x}}\right)^T, \mathbf{p}(t_f) = \left(\frac{d\phi}{d\mathbf{x}}\right)_{t=t_f} \tag{15}$$

$$\dot{R} = -\left(\frac{d\mathbf{f}}{d\mathbf{x}}\right)^T R, R(t_f) = \left(\frac{d\boldsymbol{\psi}}{d\mathbf{x}}\right)_{t=t_f} \tag{16}$$

to obtain the influence functions and a matrix of influence functions.

6. Simultaneously with the backward integration of Step 5, compute the quantities

$$I_{\psi\psi} = \int_{t_0}^{t_f} R^T \frac{\partial \mathbf{f}}{\partial \mathbf{u}} W^{-1} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)^T R dt \tag{17}$$

$$I_{J\psi} = I_{\psi J}^T = \int_{t_0}^{t_f} \left(\mathbf{p}^T \frac{\partial \mathbf{f}}{\partial \mathbf{u}} + \frac{\partial L}{\partial \mathbf{u}}\right) W^{-1} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)^T R dt \tag{18}$$

$$I_{JJ} = \int_{t_0}^{t_f} \left(\mathbf{p}^T \frac{\partial \mathbf{f}}{\partial \mathbf{u}} + \frac{\partial L}{\partial \mathbf{u}}\right) W^{-1} \left[\left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)^T \mathbf{p} + \left(\frac{\partial L}{\partial \mathbf{u}}\right)^T\right] dt \tag{19}$$

where the matrix W is an arbitrary, time varying matrix that is positive-definite.

7. Choose values of $d\boldsymbol{\psi}$ that moves the terminal condition, $\boldsymbol{\psi}[\mathbf{x}(t_f),t_f]$, closer to the desired value of $\boldsymbol{\psi}[\mathbf{x}(t_f),t_f]=0$.
8. Determine the vector

$$\mathbf{v} = -\left[I_{\psi\psi} + \frac{1}{b}\frac{d\boldsymbol{\psi}}{dt}\left(\frac{d\boldsymbol{\psi}}{dt}\right)^T\right]^{-1}\left[d\boldsymbol{\psi} + I_{\psi J} + \frac{1}{b}\left(\frac{d\phi}{dt} + L\right)\left(\frac{d\boldsymbol{\psi}}{dt}\right)\right] \tag{20}$$

where b is a weighting constant

9. Determine increments to the control vector, $\delta\mathbf{u}(t)$, and terminal time dt_f

$$\delta\mathbf{u}(t) = -W^{-1}\left[\frac{\partial L}{\partial \mathbf{u}} + \left(\mathbf{p} + R\mathbf{v}\right)^T\left(\frac{d\boldsymbol{\psi}}{dt}\right)\right] \tag{21}$$

$$dt_f = -\frac{1}{b}\left(\frac{d\phi}{dt} + \mathbf{v}^T\frac{d\boldsymbol{\psi}}{dt} + L\right)_{t=t_f} \tag{22}$$

10. Increment the estimates for the control vector, $\mathbf{u}(t)$, and the terminal time, t_f

$$\mathbf{u}^{new}(t) = \mathbf{u}^{old}(t) + \delta\mathbf{u}(t) \tag{23}$$

$$t_f^{new} = t_f^{old} + dt_f \tag{24}$$

11. Iterate using steps 4 through 10 until $\boldsymbol{\psi}\left[\mathbf{x}(t_f), t_f\right] = \mathbf{0}$, $\left[\frac{d\phi}{dt} + \mathbf{v}^T\frac{d\boldsymbol{\psi}}{dt} + L\right]_{t=t_f} = 0$, and $I_{JJ} - I_{J\psi}I_{\psi\psi}^{-1}I_{\psi J} \le \varepsilon$, where ε is the acceptable tolerance
12. Record the solution for the control history, $\mathbf{u}(t) = \mathbf{u}^{new}(t)$

This iterative solution is advantageous as it finds a local minimum in the fuel consumption robustly and as accurately as the tolerance prescribed. However, it does suffer from being computationally intensive, requiring numerous iteration before convergence occurs, particularly if a poor initial solution is given. Additionally, the algorithm is dependent on numerical derivatives which increases the number of function calls dramatically depending on the scheme used to evaluate the derivatives.

3. Closed-form, Analytic, Fuel Optimal Control Algorithm

By assuming a planar, non-rotating planet with no atmosphere, D'Souza derived an analytic, unconstrained fuel-optimal propulsive terminal descent algorithm that meets the necessary and sufficient conditions for an optimal control law[9]. The problem described by D'Souza minimizes the performance index

$$J = \Gamma t_f + \frac{1}{2}\int_{t_0}^{t_f}\mathbf{a}^T\mathbf{a}dt \tag{25}$$

which includes a weighting, Γ, on the final time. It is shown that the control law which minimizes this index, under the assumptions mentioned previously, is given by

$$\mathbf{a} = -4\frac{\Delta\mathbf{v}}{t_{go}} - 6\frac{\Delta\mathbf{r}}{t_{go}^2} - \mathbf{g} \tag{26}$$

where

$$\Delta\mathbf{r} = \begin{pmatrix} r_1 - r_{f_1} & r_2 - r_{f_2} & r_3 - r_{f_3}\end{pmatrix}^T \tag{27}$$

$$\Delta\mathbf{v} = \begin{pmatrix} v_1 - v_{f_1} & v_2 - v_{f_2} & v_3 - v_{f_3}\end{pmatrix}^T \tag{28}$$

$$\mathbf{g} = \begin{pmatrix} 0 & 0 & g \end{pmatrix}^T \tag{29}$$

The time-to-go, t_{go}, is shown from the transversality condition from the Euler-Lagrange equations to be the real positive root of the equation

$$t_{go}^4 - 2\frac{\Delta\mathbf{v}^T\Delta\mathbf{v}}{\Gamma + \frac{g^2}{2}}t_{go}^2 - 12\frac{\Delta\mathbf{v}^T\Delta\mathbf{r}}{\Gamma + \frac{g^2}{2}}t_{go} - 18\frac{\Delta\mathbf{r}^T\Delta\mathbf{r}}{\Gamma + \frac{g^2}{2}} = 0 \tag{30}$$

Equation (30) can be solved either numerically or analytically and substituted into Eq. (26) to obtain the desired acceleration vector for all time. The commanded thrust is then this acceleration vector multiplied by the mass of the vehicle at the given instant in time.

This closed-form, analytic algorithm has a clear computational advantage compared to the iterative optimal control solution as it requires a single computation for the free time-to-go which is, in turn, substituted into an equation of known state parameters (relative position and velocity to the target) to obtain the commanded thrust. However, the formation of the algorithm does not have any constraints on either the maximum thrust magnitude or minimum altitude. Without these constraints, a physically impossible solution could be obtained. However, by propagating ahead in time, violations in these constraints can be foreseen and an appropriate adjustment in the weighting on time-to-go, Γ, can be prescribed using Newton iteration. While requiring numerical integration and iteration, obtaining the proper increment on the time-to-go weighting was shown to be significantly less computationally intensive than the gradient method described previously.

4. Second-order Cone Algorithm

For propulsive terminal descent, the control space is, in general, non-convex due to a minimum allowable thrust magnitude, which none of the prior three methods described take into account. Due to this non-convex constraint, determining a control history that results in the global minimum with regards to PMF is not guaranteed. Açikmeşe and Ploen have shown that there exits a convex programming approach to the propulsive terminal descent problem which relaxes this non-convex constraint and, in turn, guarantees arrival at the global minimum in PMF[10]. Their work also reformulates the convex propulsive terminal descent guidance problem in the discretized case as a second-order cone programming problem (SOCP), which can be solved using interior-point solution methods. Interior-point methods are well studied and are known to converge to within a given tolerance of the optimum in a known, finite number of iterations, which cannot be said for any solution method of the general propulsive terminal descent guidance problem. Furthermore, the solution obtained by the interior-point method to any desired accuracy is feasible. Therefore, potential application to flight exits for this algorithm. The derivation of this algorithm assumes constant gravitational acceleration and negligible atmospheric forces; however, variations in these quantities from that modeled can be treated as disturbances when the guidance algorithm is implemented in a closed-loop sense.

The non-convex propulsive terminal descent guidance law problem can be posed as

$$\text{Minimize: } J = \int_{t_0}^{t_f} \left\|\boldsymbol{\tau}_\mathbf{C}\right\| dt$$

$$\text{Subject to: } \ddot{\mathbf{r}} = \mathbf{g} + \boldsymbol{\tau}_\mathbf{C}/m\,,\ \dot{m} = -\alpha\left\|\boldsymbol{\tau}_\mathbf{C}\right\|,\ 0 < \rho_1 < \left\|\boldsymbol{\tau}_\mathbf{C}\right\| \le \rho_2\,,\ r_3(t) \ge 0\,, \tag{31}$$

$$\left\|S_j\mathbf{x}(t) - \boldsymbol{\upsilon}_\mathbf{j}\right\| + \mathbf{c}_\mathbf{j}^T\mathbf{x}(t) + a_j \le 0,\ j = 1,\ldots,n\,, \qquad \mathbf{r}(0) = \mathbf{r}_\mathbf{0}\,, \qquad \dot{\mathbf{r}}(0) = \dot{\mathbf{r}}_\mathbf{0}\,, m(0) = m_0\,,$$
$$\mathbf{r}(t_f) = \dot{\mathbf{r}}(t_f) = \mathbf{0}$$

where convex path constraints are included. The problem can be made convex, by reformulating it in terms of a new variable, ζ. This introduces an additional constraint to ensure that the new variable acts as a slack variable.

$$\text{Minimize: } J = \int_{t_0}^{t_f} \zeta(t)dt$$

$$\text{Subject to: } \ddot{\mathbf{r}} = \mathbf{g} + \boldsymbol{\tau}_\mathbf{C}/m\,,\ \dot{m} = -\alpha\left\|\boldsymbol{\tau}_\mathbf{C}\right\|,\ \left\|\boldsymbol{\tau}_\mathbf{C}\right\| \le \zeta(t)\,,\ 0 < \rho_1 < \zeta(t) \le \rho_2\,,\ r_3(t) \ge 0 \tag{32}$$

$$\left\|S_j\mathbf{x}(t) - \boldsymbol{\upsilon}_\mathbf{j}\right\| + \mathbf{c}_\mathbf{j}^T\mathbf{x}(t) + a_j \le 0,\ j = 1,...,n\,, \qquad \mathbf{r}(0) = \mathbf{r}_\mathbf{0}\,, \dot{\mathbf{r}}(0) = \dot{\mathbf{r}}_\mathbf{0}\,, m(0) = m_0\,,$$

$$\mathbf{r}(t_f) = \dot{\mathbf{r}}(t_f) = \mathbf{0}$$

Graphically, this transformation of the non-convex control space to convex control space by the introduction of this slack variable is shown in Figure 17 for a two-dimensional case. In general, the solution to the non-convex problem is a feasible solution to the relaxed problem; however, the converse is not guaranteed to be true. However, Açikmeşe and Ploen have shown that the optimal solution found by the relaxed problem is also the optimal solution to the non-convex problem[10].

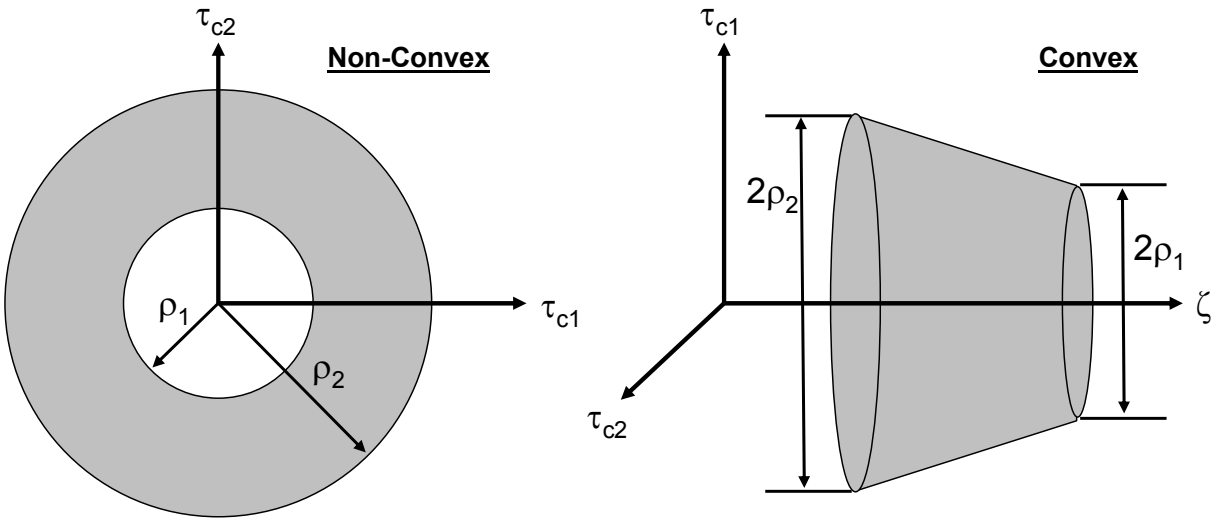


Figure 17. Non-convex thrust control space and convex thrust control space.

To implement this algorithm numerically, the continuous time problem needs to discretized. In their work, Açikmeşe and Ploen describe a change of variables transformation for Eq. (32) that leads to a continuous optimization problem, but one which has convex performance index as well as convex state and control constraints. In particular, the change of variables they introduce lead to constraints that are either linear or of the form of a second-order cone. Following this change of variables, the problem is discretized in time while enforcing the constraints at the nodes of the resulting mesh which results in a finite-dimensional SOCP problem. The resulting SOCP can be solved using an interior-point method algorithm which will arrive at the global minima in polynomial time[10].

Remark: When implementing this algorithm, several existing packages exist. In particular, SeDuMi was utilized for this study, which is a Matlab oriented software package that solves symmetric cone optimization problems, such as the problem posed here[11]. Should one not want to rely on a preexisting package, several other SOCP solution methods exist such as the interior-point method described by Lobo, *et al*, and the Q method for second-order cones described by Xia and Alizadeh[12,13].

5. Comparison of the Four Propulsion Guidance Algorithms

The four propulsive terminal descent algorithms were evaluated in a dispersed environment shown in Table 14 for the case of a large robotic entry vehicle from the parachute deployment point through the soft touchdown assuming ideal navigation knowledge. One-thousand cases for each algorithm were sampled and the PMF required to achieve pinpoint accuracy was computed. Figure 18 shows the cumulative distribution functions (CDFs) for the PMF of the gradient based optimal guidance algorithm, the closed-form analytic guidance algorithm, and the modified Apollo lunar module guidance algorithm. The CDF of the second-order cone algorithm is plotted separately in Figure 19 due to there being a minimum thrust bound (1142±57 N) that is not imposed on the others. Results of a smaller, 100 case Monte Carlo analysis shows that the second-order cone guidance law behaves similarly to that of the optimal gradient based algorithm when the lower thrust bound constraint is eliminated. Therefore, in addition to the second-order cone guidance law, Figure 19 also shows the CDF from the optimal gradient based algorithm, which was more completely sampled, to show the effective cost of this minimum thrust constraint. Table 15 provides a quantitative and qualitative comparison between each of the four algorithms, where the qualitative metrics were assessed on a relative basis to one another.

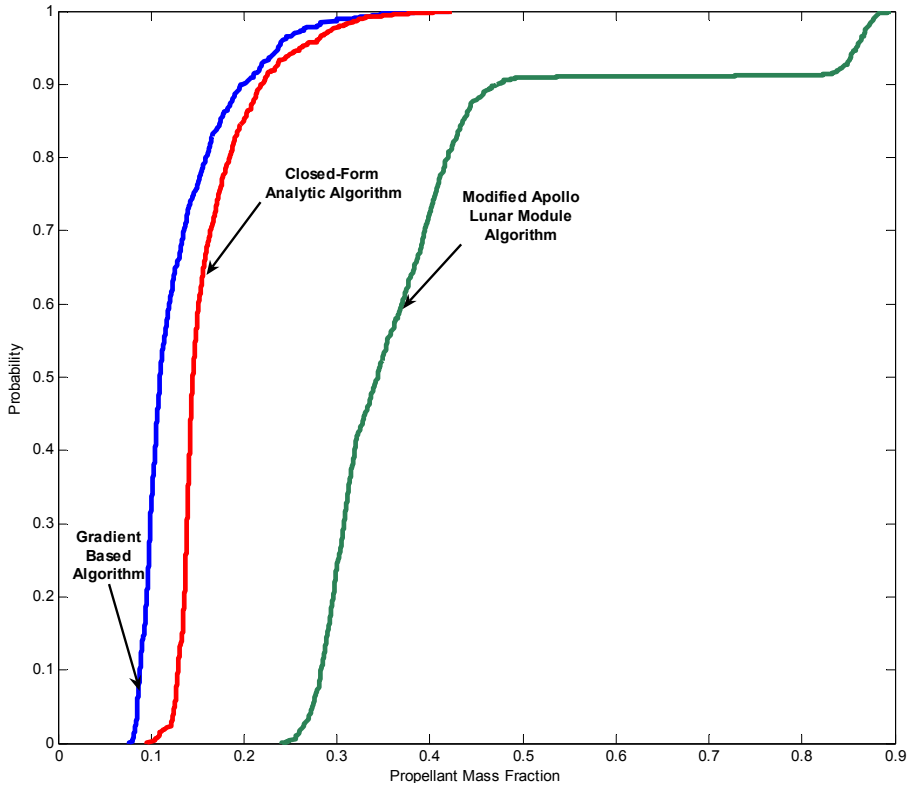


Figure 18. CDFs of the PMFs for the three algorithms without a minimum thrust constraint.

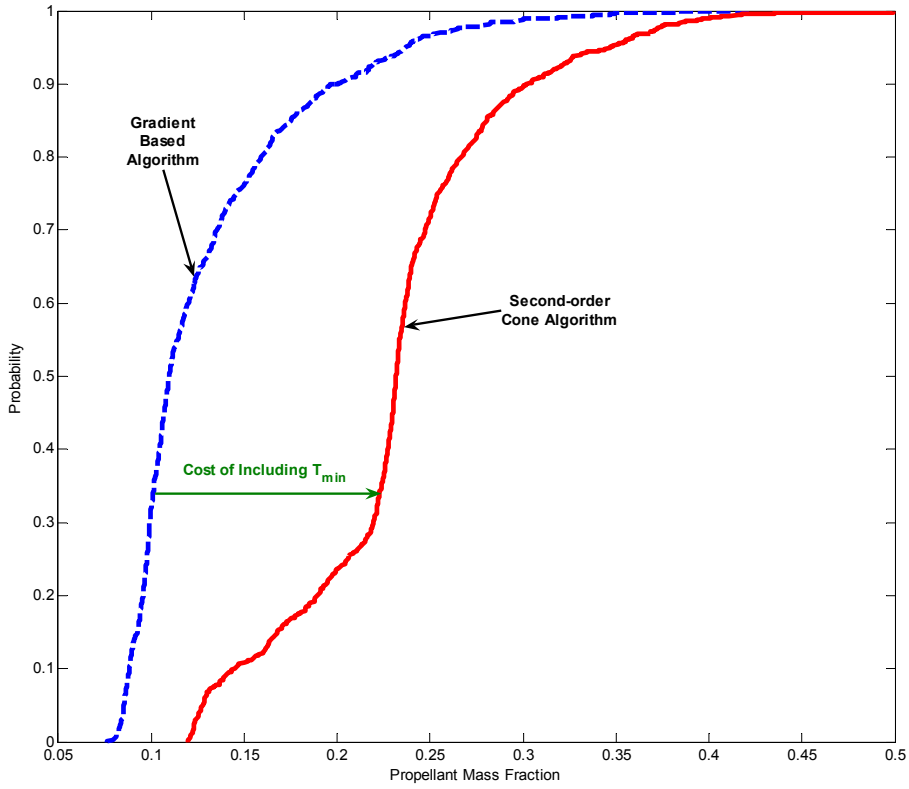


Figure 19. CDF of the PMF for the second-order cone algorithm with the cost of the minimum thrust constraint.

Table 15. Comparison of the four propulsive descent algorithms.

	Modified Apollo Lunar Module Algorithm	*Gradient Based Optimal Control Algorithm*	*Closed-Form, Analytic Algorithm*	*Second-order Cone Algorithm*
Optimality at 99% C.I. [PMF_{min}/PMF]	0.43	1.00	0.99	0.93
Equations of Motion Evalutations	1	327	8	163
Robustness	Poor	Good	Moderate	Good
Ease of Implementation	Good	Poor	Good	Moderate
Applicability to Flight	Good	Moderate	Moderate	Good
Numerical Stability	Good	Moderate	Moderate	Good

At the 99% confidence level, there is little discernable difference in the PMF for the gradient based optimal control law or the closed-form analytical higher for the computational advantageous closed-form analytic algorithm. Even at lower confidence levels, the PMF is only a few percent that the PMF for the modified Apollo lunar module algorithm is significantly higher. In fact, the PMF is unachievably high for the higher confidence levels. This is due to the lack of robustness to the environmental parameters variations seen as the algorithm was developed for use on a planetary body lacking an atmosphere and the prediction of the acceleration profiles don't agree with that experienced. Therefore, in order to achieve a soft landing with pinpoint accuracy, a constant altitude translational maneuver is undertaken once the vehicle has crossed an altitude threshold (200 m AGL).

For application in conceptual design studies, the PMF performance of the closed-form analytic algorithm is sufficiently close to the gradient based fuel-optimal PMF. As an example of the computational efficiency of the closed-form algorithm as compared to the iterative gradient algorithm, for a single control history determination (i.e., a call of the closed-loop guidance algorithm at one instant in time during the descent), the closed-form algorithm required eight function evaluations of the equations of motion whereas the iterative algorithm required 327 function evaluations of the equations of motion. This is a clear computational advantages afforded by the closed-form analytic algorithm and makes this algorithm the preferred algorithm for further studies conducted.

However, for flight applications, the second-order cone algorithm is preferred. As observed in Figure 19, the PMF for this algorithm is higher than that of the optimal gradient based algorithm. This is a result of providing a realistic minimum thrust magnitude constraint in addition to the maximum thrust magnitude constraint imposed for the algorithms. For flight systems, the cycling of engines on and off, which can result from not specifying a minimum thrust constraint, is not desired as it increases the probability of loss of mission. A smaller sampling of the dispersions showed that the second-order cone and the optimal gradient based algorithm behave similarly, which is to be expected as the gradient based algorithm, without the minimum thrust constraint, is convex. Therefore, both the gradient based algorithm and the second-order cone algorithm should arrive at the same PMF for the propulsive pinpoint landing problem. As seen in Table 15, the number evaluations of the equations of motion for the second-order cone algorithm is half that of the gradient based algorithm, this results in a run-time on the order of one-quarter that of the gradient based algorithm due to the polynomial convergence properties of the SOCP, making the SOCP preferable in conceptual studies as well to the gradient based optimal guidance law.

D. Subsonic Guided Parachute Performance

A subsonic steerable parachute was included in the entry vehicle system used for terminal descent. Several other studies have investigated the inclusion of subsonic parachutes in the past, these include one performed by Mitcheltree, *et al*, in which a test program is laid out for Earth qualification of the parachute and one performed by Witkowski, *et al*, where the inclusion of a subsonic chute was investigated[14,15]. For this trade, a circular parachute with a nominal drag coefficient of 0.68 was assumed. A guidance algorithm model based on that described by Yakimenko, *et al*, which modulates the drag vector in order to steer towards a reference trajectory using a performance index to minimize the amount of time it takes to arrive on the reference[16]. For evaluation purposes, the parachute is deployed at Mach 0.9 in the descent trajectory subsequent to release of the supersonic parachute and is used until 1 km AGL. At this altitude, the analytic propulsive guidance algorithm is activated. One thousand Monte Carlo cases were conducted using the parameters in Table 14.

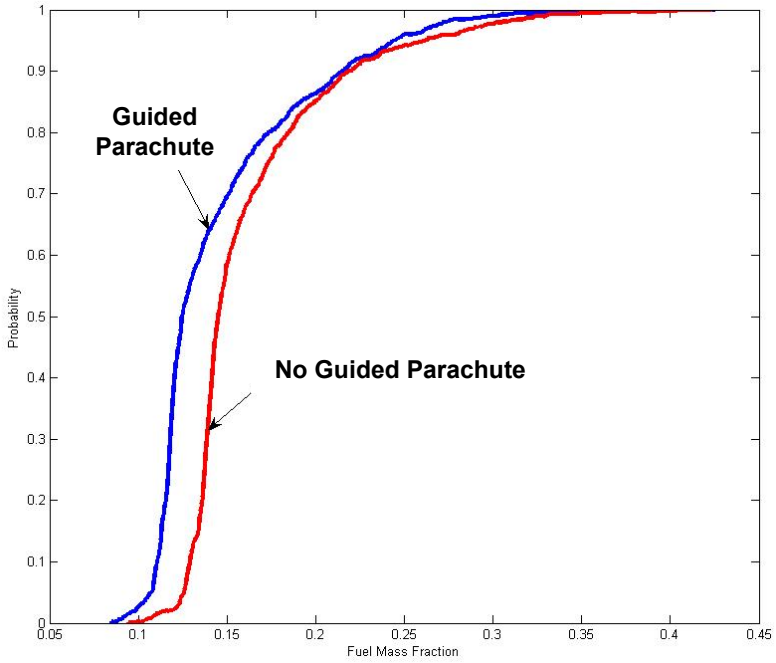


Figure 20. CDFs of the benefit of a subsonic guided parachute.

As shown in Figure 20, the results of this Monte Carlo analysis show no major improvement in the PMF required to achieve pinpoint accuracy. The principal reason for this lack of benefit is that the timeline associated with a significant fraction of the Martian entries (*e.g.*, a CDF above 80%) does not allow for the subsonic guided parachute to be inflated for a significant length of time, as would occur on in Earth

applications. For all of the cases investigated the deployment of the subsonic parachute occurred below 5 km allowing a maximum 4 km guided descent segment. Additionally, there is no consideration for the energy state once on the trajectory, which could be a dominant factor as a spiraling trajectory towards the reference will increase the energy and increase the propulsive force required to negate it.

E. Hypersonic Guidance Performance

The ramifications of the inclusion of a hypersonic guidance algorithm was evaluated by examining the semi-major axis of the supersonic parachute deployment ellipse and examining the PMF required to achieve a pinpoint landing using the closed-form, analytic guidance law and ideal navigation. As shown in Figure 21, these results demonstrate a marginal PMF reduction below a position delivery error at parachute deployment of 3 km. Because a supersonic parachute deployment ellipse greater than 3 km leads to a dramatic rise in the PMF required to achieve pinpoint landing, a 3 km dispersion footprint is suggested as a target for Mars hypersonic guidance algorithm technology development.

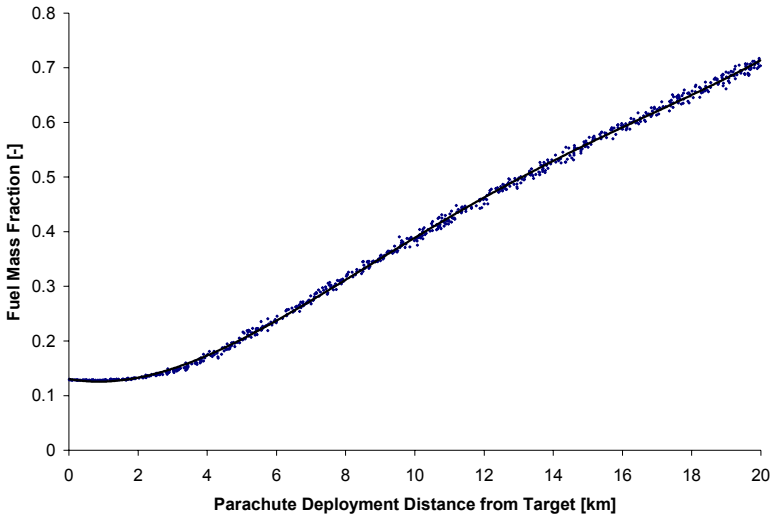


Figure 21. PMF for supersonic parachute deployment semi-major errors.

F. Terrain Relative Navigation Sensor Performance

The impact of terrain relative navigation on the ability to perform pinpoint landing was studied by first assuming a baseline sensor suite consisting of an inertial measurement unit consisting of an accelerometer and gyroscope, a radar, a velocimeter, and a 3-dimensional TRN sensor. This baseline sensor suite has the specifications shown in Table 16. Additionally, a covariance matrix at EI-10 minutes was obtained from the Jet Propulsion Laboratory for MSL assuming TCM-5 was performed. This entry uncertainty was propagated to the parachute deployment ellipse with sensor error accumulating throughout the descent. Using this baseline sensor suite, a navigational knowledge error of approximately 200 m was seen at parachute deployment. Thus, with the baseline sensor suite, pinpoint landing is precluded.

Table 16. Baseline navigation sensor data.

Sensor	*Altitude Active [km]*	*Termination Altitude [km]*	*Bias*	*Sensor Realization Error*	*Noise (1σ)*
Accelerometer	N/A	N/A	30 μg	66 ppm	5 $\mu g/Hz^{1/2}$
Gyroscope	N/A	N/A	0.02 deg/hr	1.6 ppm	50 $\mu rad/Hz^{1/2}$
Radar	20	0.01	-0.1 m	0 m	5 m
Velocimeter	2	0.01	(-0.14, 0.07, -0.06) m/s	(5.79×10-5, 5.79×10-5, 5.79×10-5) rad	0.5 m/s
3D Terrain Relative Navigation Sensor	8	2.5	(4.3, 28.7, -15.7) m	(0, 0, 0) m	42 m

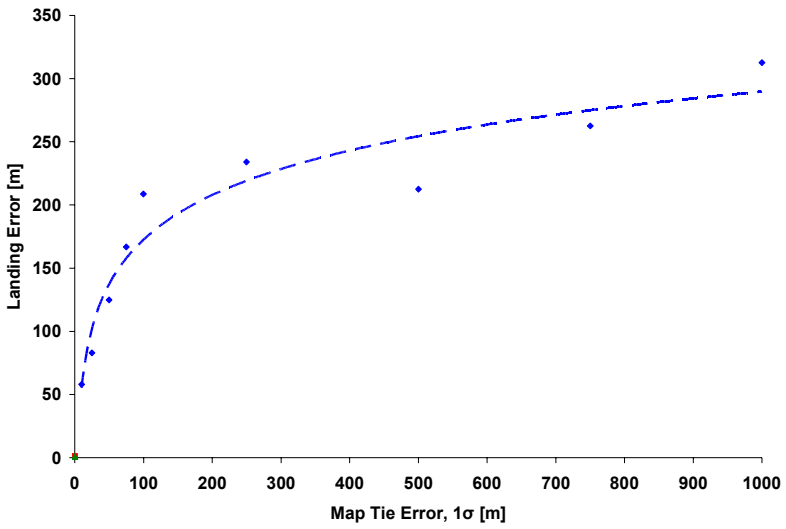


Figure 23. Map-tie error impact on landing uncertainty.

By varying the termination altitude of the TRN sensor, a surrogate of the performance achievable by the sensor is available. Additionally, the frequency of navigation data updates performed inflight is investigated. The landed error variation for the nominal case is seen in Figure 22 as a function of these two variables. As expected, the general trend show improving landed accuracy with decreased termination altitude. A minimum is seen as well for the 0.1 Hz and 0.5 Hz case at approximately 1.5 km. Sensor data below this altitude may be more inaccurate that the known state, which, in turn, leads to the overall knowledge error decreasing. This is not seen in the 1 Hz sampling rate as more accurate data is obtained between 1 and 1.5 km improving the knowledge of the known state.

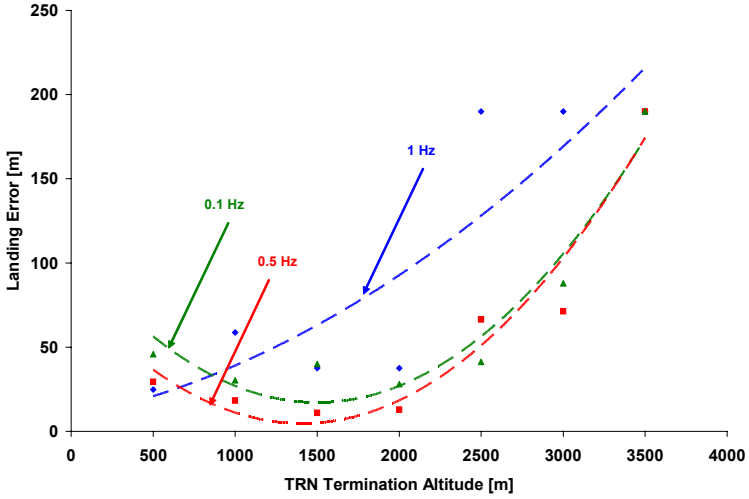


Figure 22. Landed accuracy for various TRN termination altitudes and sampling frequencies.

Additionally, the impact of map-tie error was investigated by considering values from 0 to 1000 m and examining the total laded accuracy of the vehicle. As shown in Figure 23, for a sub-100 m landed accuracy, the maximum allowable one standard deviation map-tie error is approximately 25 m. The variation in landed uncertainty with map-tie error is approximately logarithmic. This can be attributed to the TRN

sensor negating the majority of the map-tie error above 200 m and is largely a function of the capabilities of the sensor.

G. Conclusions

This investigation compared the performance of four propulsive terminal descent algorithms for a sub-100 m landing accuracy application at Mars. An iterative propellant optimal guidance law was implemented. However, it was seen that a near-optimal guidance law which assumes a flat, atmosphere-free planet was of sufficient accuracy for conceptual design, while being significantly less computationally intensive. Additionally, a convex algorithm which arrives at the constrained global minimum in PMF was investigated for its strong applicability to flight. Pinpoint landing technology system trade studies performed demonstrated that driving hypersonic guidance requirements to an accuracy below 3 km results in marginal performance gains. In addition, it was shown that the use of a subsonic guided parachute does not significantly decrease the propellant mass fraction required for pinpoint landing, while the added system complexity of a subsonic guided parachute would have to be seriously considered. This investigation also demonstrates that navigational uncertainty is the principal driving factor precluding pinpoint landing at Mars with map-tie error needing to be driven below 25 m and TRN sensor termination altitude needing to be driven below 1 km for sub-100 m landed accuracy.

The results of this study were obtained assuming a parachute deployment ellipse with similar semi-major axis length to MSL at a constant altitude. Accounting for the altitude variations that will inevitably result from the system and environment dispersions should be accounted for in order to further add fidelity to these results. The initiation of the propulsive descent for this work started at Mach 0.9; however, a trade between the PMF and the initiation state relative to the target exists and should be explored in future work. A third area of potential extension of this work is the influence of various hypersonic phase guidance algorithms (*e.g.*, a predictor-corrector algorithm or the modified Apollo algorithm) and navigation performance on the propulsive terminal descent phase. For instance, does a predictor-corrector algorithm that targets an optimal parachute deployment altitude for the terminal descent guidance algorithm strongly impact the performance of the system?

H. References

[1]Wolf, A., Tooley, J., Ploen, S., Gromov, K., Ivanov, M., and Acikmese, B., "Performance Trades for Mars Pinpoint Landing," *2006 IEEE Aerospace Conference*, Paper 1661, Big Sky, Montana, March 2006.

[2]Braun, R.D. and Manning, R.M., "Mars Exploration Entry, Descent, and Landing Challenges," *Journal of Spacecraft and Rockets*, Vol. 44, No. 2, pp. 310-323, 2007.

[3]Mendeck, G.F., and Carman, G.L., "Guidance Design for Mars Smart Landers Using the Entry Terminal Point Controller," Paper No. AIAA 2002-4502, AIAA Atmospheric Flight Mechanics Conference, 5-8 August 2002, Monterey, California.

[4]Drake, B. G. (Ed.), "Reference Mission Version 3.0 Addendum to the Human Exploration of Mars: The Reference Mission of the NASA Mars Exploration Study Team," NASA-SP-6107-ADD, Jun. 1998.

[5]Striepe, S., *et al*, "Mars Science Laboratory Simulations for Entry, Descent, and Landing," *Journal of Spacecraft and Rockets*, Vol. 43, No. 2, pp. 311-323, 2006.

[6]Duvall, A., *et al*, "Global Reference Atmospheric Model (GRAM) Series for Aeroassist Applications," Paper No. AIAA 2005-1239, AIAA Aerospace Sciences Meeting and Exhibit, January 2005, Reno, Nevada.

[7]Wong, E., Singh, G., and Masciarelli, J., "Autonomous Guidance and Control Design for Hazard Avoidance and Safe Landing on Mars," AIAA Atmospheric Flight Mechanics Conference and Exhibit, August 2002, Monterey, California.

[8]Bryson, A., and Ho, Y., *Applied Optimal Control*, Hemisphere Publishing Company, 1963.

[9]D'Souza, C., "An Optimal Guidance Law for Planetary Landing." Paper No. AIAA 97-3709, 1997.

[10] Açikmeşe, B., and Ploen, S., "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, pp. 1353-1366, 2007.

[11] Sturm, J., "Using SeDuMi 1.02, a MATLAB Toolbox for Optimization Over Symmetric Cones," *Optimization Methods and Software*, Vol. 11, No. 1, pp. 625-653, 1999.

[12] Lobo, M., et al, "Applications of Second-Order Cone Programming," *Linear Algebra and its Applications*, Vol. 284, No. 1, pp. 193-228, 1998.

[13] Xia, Y., and Alizadeh, F., "The Q method for second order cone programming," *Computers & Operations Research*, Vol. 35, pp. 1510-1538, 2008.

[14]Mitcheltree, R., *et al*, "High Altitude Test Program for a Mars Subsonic Parachute", *18th AIAA Aerodynamic Decelerator Conference*, AIAA 2005-1659, Munich, Germany, May 2005.

[15]Witkowski, A., Machalick, A., and Taeger, Y., "Mars Subsonic Parachute Technology Task System Overview", *18th AIAA Aerodynamic Decelerator Conference*, AIAA 2005-1657, Munich, Germany, May 2005.
[16]Yakimenko, O., *et al*, "On Control of Autonomous Circular Parachutes," Paper No. 2002-4753, AIAA Atmospheric Flight Mechanics Conference, 5-8 August 2002, Monterey, California.

III. Smart Divert: A New Entry, Descent, and Landing Architecture

Smart Divert: A New Entry, Descent, and Landing Architecture

Michael J. Grant[2] and Robert D. Braun[3]
Georgia Institute of Technology, Atlanta, GA, Zip, 30332

Mars robotic landing site selection has been a compromise between scientific interest and safety. Due to the rather large landed footprint major axis lengths of Viking, Pathfinder, Mars Exploration Rovers, and Phoenix, mission designers have been forced to orient the landed ellipse in vast, relatively flat areas to provide high probability of landing success. Scientists are interested in exploring more geologically interesting areas that contain many hazards, including sloping terrain, craters, and rocks. Smart Divert provides a new entry, descent, and landing architecture that could allow robotic missions to safely land in hazardous terrain. Smart Divert consists of a ballistic entry followed by supersonic parachute deployment. After parachute release, the vehicle diverts to one of many predefined, fuel-optimal safe zones. Smart Divert performance and entry design is discussed and is followed by a discussion of Smart Divert for random terrain. An initial assessment of optimal landing site arrangement is performed and an example of the usefulness of Smart Divert is performed for real terrain using Phoenix landing site rock count data.

Nomenclature

$\underline{a}$	=	Acceleration vector
$\underline{g}$	=	Local gravity vector
J	=	Cost function
t_0	=	Time of divert initiation
t_f	=	Final touchdown time
t_{go}	=	Time-to-go
Γ	=	Weighting on time-to-go
$\underline{\Delta r}$	=	Position of vehicle relative to target
$\underline{\Delta v}$	=	Velocity of vehicle relative to target
AGL	=	Above ground level
DOF	=	Degree-of-freedom
DDOR	=	Delta Differential One-Way Ranging
DGB	=	Disk-gap-band
EDL	=	Entry, descent, and landing
HiRISE	=	High Resolution Imaging Science Experiment
JPL	=	Jet Propulsion Laboratory
MER	=	Mars Exploration Rovers
MRO	=	Mars Reconnaissance Orbiter
MSL	=	Mars Science Laboratory

[2] Graduate Research Assistant, Guggenheim School of Aerospace Engineering, AIAA Student Member.
[3] David and Andrew Lewis Associate Professor of Space Technology, Guggenheim School of Aerospace Engineering, AIAA Fellow.

PMF = Propellant mass fraction
RCS = Reaction control system
TCM = Trajectory correction maneuver

A. Introduction

TO date, entry, descent, and landing (EDL) mission designers have been forced to trade safety and scientific interest when choosing the landing site of various Mars landers. Past missions have resulted in rather large landed footprint major axis lengths ranging from 200 km (Viking, Pathfinder) to 80 km (Mars Exploration Rovers).[1] Generally, scientifically interesting landing sites are not flat and contain many hazards including significant variation in terrain elevation, craters, and rocks. Hence, it is the interest of scientists to visit these dangerous regions on Mars. However, mission designers must ensure that the majority of the landed ellipse encapsulates safe terrain. This leads the mission designers to orient the landing site ellipse over vast flat regions of Mars. It would be very beneficial if science could dominate the choice of landing site location. This could be achieved with greater landed accuracy. In order to achieve improved accuracy, hypersonic guidance will be used for the first time at Mars on the Mars Science Laboratory (MSL). The modified Apollo guidance utilizes only the terminal phase of the Apollo entry guidance and provides a landed footprint 20 km long in major axis through modulation of the lift vector.[2] This allows MSL to travel to more dangerous and scientifically interesting sites. However, the implementation of hypersonic guidance greatly increases complexity and cost of the mission as compared to previous ballistic, unguided missions. Also, mission designers are still required to ensure the majority of the 20 km landed ellipse is over safe terrain. Smart Divert may provide a simple, low cost entry, descent, and landing architecture for landing in dangerous terrain. Smart Divert consists of a ballistic entry followed by supersonic disk-gap-bank (DGB) parachute deployment at Mach 2.2, consistent with previous missions. At Mach 0.8, the parachute is released and the vehicle propulsively diverts to a fuel-optimal safe zone identified prior to EDL.

B. Planar Example

The visualization of 3D divert trajectories is quite difficult. In order to conceptually understand Smart Divert, a simple planar example was constructed. The entry flight path angle was varied to produce a ballistic parachute deploy line (all trajectories are confined to a plane) as shown in a and b of Figure 24. Note that the downrange spread at parachute deploy is only approximately 10 km. This is unrealistically small and is only used to conceptually demonstrate Smart Divert. In Figure 24a, a single site is located at 0 km downrange. After the ballistic entry and parachute phase, each trajectory diverts to the target site on a fuel optimal trajectory. The diverts that initiate far uprange must divert a long distance, requiring more fuel than the diverts that initiate closer to the target. In order to reduce the propellant mass fraction (PMF)

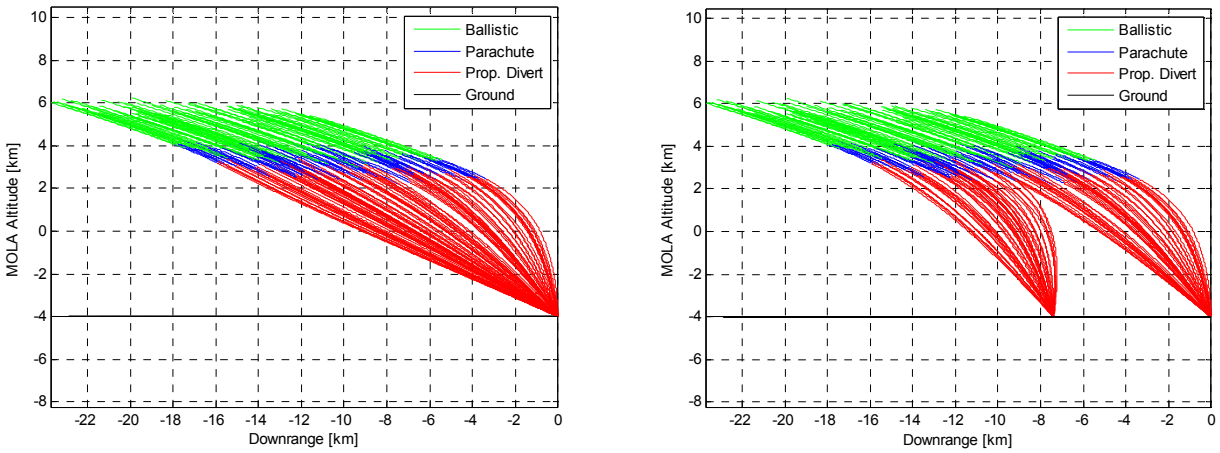


Figure 24: Example Divert to One and Two Sites

required by the uprange trajectories, a second site was added uprange and may be seen in Figure 24b. As can be seen, the vehicle evaluates which site is fuel optimal and flies to that site. Hence, the uprange trajectories identified the uprange site as fuel optimal and diverted to that site.

C. Simulation Development

In order to have a flexible conceptual design tool that is capable of rapidly trading various EDL scenarios, a 3 degree-of-freedom (DOF) Mars entry simulation was developed in Matlab. The Matlab code was autocoded into a C-Mex file using the Matlab Real-Time Workshop, which dramatically reduces the execution time by an approximate factor of 35. The equations of motion were expressed in an inertial, Cartesian space. This method avoids singularities associated with angular derivatives (*e.g.*, rate of change in latitude, longitude, flight path angle, etc) as the vehicle's velocity approaches zero (*e.g.*, during terminal descent). A spherical, rotating planet with a spherical mass distribution was also assumed.

In order to assess the performance of Smart Divert, a Monte Carlo environment was developed with dispersions similar to those simulated for MSL.[3] Atmospheric properties, vehicle properties, parachute aerodynamic drag coefficient, and delivery accuracy to Mars are dispersed and are shown in Table 18 and Table 17. An MSL-class DGB parachute with a diameter of 19.5 m and drag coefficient profile shown in Figure 25 was used. Note the drag bucket near Mach 1 is captured and reduces parachute performance. This will be an important consideration for low parachute deployment altitudes performed in subsequent analyses. The delivery accuracy was quantified as an entry state covariance at ten minutes prior to entry interface provided by the Jet Propulsion Laboratory (JPL) for MSL assuming the trajectory correction maneuver (TCM) 5 was performed. This covariance corresponds to state-of-the-art interplanetary navigation capability in which the vehicle is spin stabilized, delta differential one-way ranging

Table 18: MarsGRAM 2005 Parameters

Parameter	Value / Range
Latitude [deg]	-40.60
Longitude [deg]	-62.90
Date	26 Jul 2010
Dusttau	0.1-0.9

Table 17: Monte Carlo Dispersions

Parameter	Nominal	Distribution	3σ or min/max
Entry State	MSL Nominal	Entry Covariance	Entry Covariance
C_a Multiplier	1	Gaussian	3%
Entry Mass [kg]	2196.0	Gaussian	2.0
Atmosphere Dispersion Seed	0	Uniform	1/29999
Atmosphere Update Distance [km]	0.5	Uniform	0.5/5.0
Dusttau	0.45	Uniform	0.1/0.9
Supersonic Parachute Drag	C_D Profile	Uniform	-10%/+10%
Terminal Descent Engine I_{sp} [sec]	194	Uniform	-0.67%/+0.67%

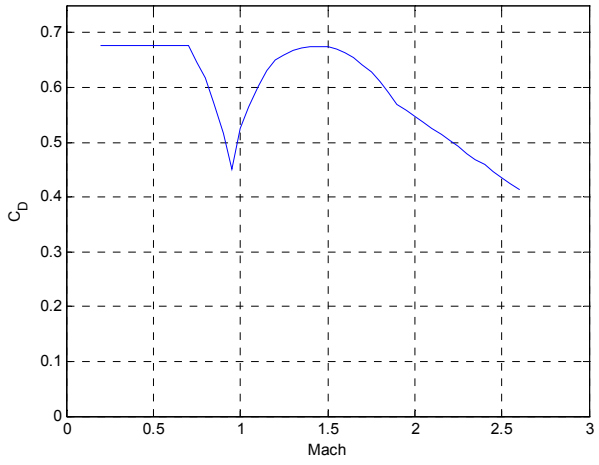


Figure 25: Drag Coefficient vs. Mach for DGB Parachute

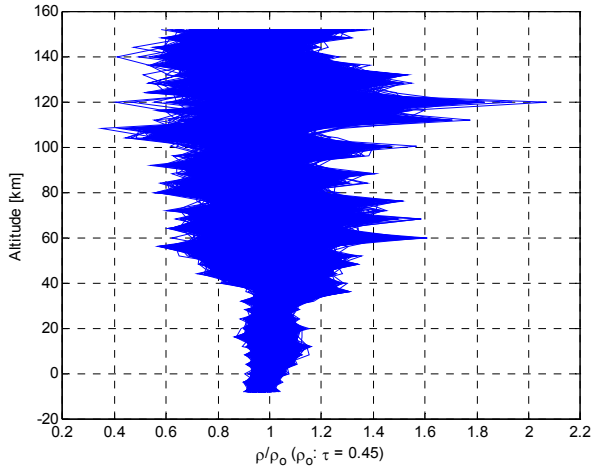


Figure 26: Atmosphere Density Profiles

(DDOR) is used, and a delivery error reducing TCM 5 is performed. The corresponding MSL 3σ entry flight path angle uncertainty is approximately 0.1^{o} as opposed to the Phoenix 3σ entry flight path angle uncertainty of 0.21^{o}.

A set of dispersed atmospheres was generated using MarsGRAM 2005. The parameters used to generate the atmospheres are shown in Table 17. The resulting atmosphere density profiles, normalized by the nominal density profile with a Dusttau of 0.45, may be seen in Figure 26. As expected, large variations occur in the upper atmosphere. Perfect navigation throughout the EDL phase was also assumed.

D. Divert Guidance

At Mach 0.8, the parachute is released and the propulsive terminal descent phase is initiated in which the vehicle diverts from its current location to the fuel optimal safe zone. The identification of the fuel optimal safe zone could be accomplished in two ways. First, a guidance algorithm could be used along with simplified equations of motion propagated onboard the vehicle. The fuel optimal safe zone could then be chosen autonomously after evaluating the propellant required to divert to each possible safe zone. This method was employed for this analysis. Alternatively, the selection of the divert site could be pre-selected by the ground. Due to the simplicity of ballistic entries, the distance traveled downrange could possibly be inferred by the entry acceleration profile. For example, the peak deceleration loads could potentially be used to identify where along the major axis the vehicle is likely located. From this estimated location, the vehicle could then divert to the corresponding fuel optimal safe zone identified by previous analysis on the ground, eliminating the need for complicated, real-time, onboard analysis.

A closed-form, analytic, fuel optimal control algorithm (D'Souza guidance) has been identified as a fuel optimal terminal descent control law for conceptual design.[4,5] The algorithm assumes a planar, non-rotating planet with negligible atmospheric forces compared to gravity and thrust. The altitude is also assumed to be much smaller than the radius of the planetary body. These assumptions are quite reasonable during terminal descent where the vehicle is close to the ground and traveling at subsonic speeds. The analytic D'Souza guidance provides a fuel-optimal, propulsive control law to perform the divert maneuver from the current time, t_0, to the final touchdown time, t_f, by minimizing the performance index shown in Eq. (1) where the weighting on time-to-go, Γ, is set initially to zero. The analytic control law is shown to be given by Eq (2), where the time to go, t_{go}, is the real positive root of Eq (3). $\underline{\Delta r}$ and $\underline{\Delta v}$ correspond to the relative position and velocity of the vehicle with respect to the target, respectively, as defined by D'Souza.[5] $\underline{a}$ corresponds to the vehicle acceleration vector, and $\underline{g}$ corresponds to the local gravity vector. The required thrust vector may then be easily obtained from the vehicle's current mass.

$$J = \Gamma t_f + \frac{1}{2}\int_{t_0}^{t_f} \left(\underline{a}^T \underline{a}\right) d\tau \tag{1}$$

$$\underline{a} = \frac{-4\underline{\Delta v}}{t_{go}} - \frac{6\underline{\Delta r}}{t_{go}^2} - \underline{g} \tag{2}$$

$$\left(\Gamma - \frac{g^2}{2}\right) t_{go}^4 - \left(2\underline{\Delta v}^T \underline{\Delta v}\right) t_{go}^2 - \left(12\underline{\Delta v}^T \underline{\Delta r}\right) t_{go} - 18\underline{\Delta r}^T \underline{\Delta r} = 0 \tag{3}$$

Immediately prior to initiating the divert, the vehicle evaluates the fuel optimality of each safe zone by propagating simplified equations of motion using the D'Souza guidance. Certain fuel optimal trajectories go through the surface of the planet. If this occurs during the evaluation of a trajectory to a specified safe zone, Γ is increased until a feasible trajectory that remains above the surface is found. An increase in Γ

results in an increase weighting on the final time, resulting in more direct trajectories that remain above the surface at the penalty of increased fuel consumption.

The analytic nature of the control law is computationally inexpensive (relative to other guidance algorithms) and allows for rapid execution of Monte Carlos. A maximum thrust to weight ratio of 3 was used for the propulsive terminal descent (consistent with historical Mars robotic monopropellant hydrazine terminal descent systems).[6] Consequently, the thrust was limited if the analytic D'Souza algorithm commanded more thrust than permitted by the thrust to weight constraint. Navigation is assumed to be perfect throughout EDL. Using the D'Souza guidance algorithm and perfect EDL navigation, the miss distance of the vehicle at touchdown to the target is approximately 5 m. Hence, the landed accuracy of the vehicle is governed by navigation error. It has been shown that in order to achieve pinpoint landing accuracy (sub-100 m), terrain-relative navigation and a reduction in map-tie error will be required.[4] For Smart Divert, the landed accuracy dictates how large the safe zones must be to ensure a safe landing.

E. Conceptual Understanding of Smart Divert Performance

For dispersed trajectories, the flight path angle, altitude, and divert distance will vary. However, it is important to gain an understanding of the reasonable bounds of Smart Divert. For any given dispersed trajectory, the fuel-optimal divert site may be located along or against the natural direction of motion of the vehicle. As an average for this analysis, the vehicle is assumed to be traveling vertically downward at Mach 0.8 (immediately after parachute jettison). The altitude above the ground in which the divert is initiated was varied from 4-12 km and the horizontal distance of the divert was varied from 0-50 km. The resulting PMF for the various combinations of divert initiation altitudes and horizontal divert distances may be seen in Figure 27. The white region corresponding to the altitude of divert initiation between 4-12 km indicates divert trajectories that require a thrust-to-weight ratio larger than 3 or maximum Mach values larger than 0.8. Such cases are considered infeasible when performing a propulsive divert in a landing configuration. As expected, an increase in horizontal divert distance requires a higher divert initiation altitude. This ratio provides an effective glide slope of 3:1 for the divert. It is also important to note that initiating the divert at a higher altitude slightly increases the required PMF for the same horizontal divert distance. Thus, the vehicle should initiate the divert at as low of an altitude as possible.

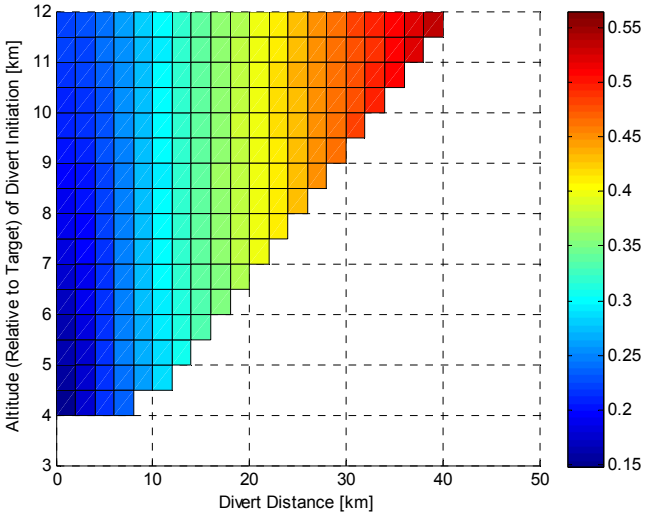


Figure 27: Propellant Mass Fraction for Various Diverts

In order to feasibly implement Smart Divert as a new EDL architecture, the propellant required to perform the required diverts must be maintained at a reasonable level. Previous EDL missions that require only a safe landing on the surface of Mars typically employ a variation of a gravity turn. The PMF required to perform a gravity turn is approximately 0.15. This is consistent with the 4 km altitude divert of 0 km. In order to feasibly implement Smart Divert, the propellant required should stay small and not double the required propellant to perform a gravity turn. Therefore, the vehicle should not divert more than 10 km to ensure the PMF required to perform the divert is less than 0.3. Since the vehicle should start the divert as low as possible, the vehicle should start the divert around 5 km above ground level (AGL) to divert a maximum of 10 km. This would allow sufficient timeline to perform the divert and other final EDL events.

F. Entry Design

The entry covariance obtained from JPL corresponds to the MSL mission with TCM 5 performed. The nominal relative entry flight path angle is -15.7° for this covariance. Since MSL is a lifting entry, such a steep entry flight path angle results in a relatively high parachute deploy altitude. The ballistic entry of Smart Divert will not result in such high parachute deploy altitudes for steep entries. In order to increase altitude for ballistic entries of a given system, the entry flight path angle must be shallowed. Figure 28 shows the impact of the nominal entry flight path angle on parachute deploy MOLA altitude and landing ellipse length where each box represents a 1° change in EFPA from the nominal MSL value of -15.7°. As expected, the parachute deploy altitude decreases as the entry flight path angle is steepened. As the nominal entry flight path angle is changed, the error in entry flight path angle remains unchanged and is the same as the entry flight path angle error obtained from the original MSL covariance. Hence, the influence of the error in entry flight path angle on the landing ellipse length increases as the entry flight path angle is shallowed. This is extremely important for mission design. A shallower entry flight path angle permits high parachute deploy altitudes at the cost of an increase in landing ellipse length. Thus, in order to perform Smart Divert at high elevations, the corresponding landing ellipse length will be quite large. For the shallowest entry near -11°, the corresponding landing ellipse length is approximately 80 km, consistent with MER. Consequently, a great number of safe zones will have to be identified for high altitude parachute deploy conditions to ensure the required PMF remains reasonable. In order to provide a 5 km spread between terminal descent initiation and the ground, it is unlikely that landing site elevations greater than 2 km MOLA will be chosen for Smart Divert unless a large number of safe zones can be identified.

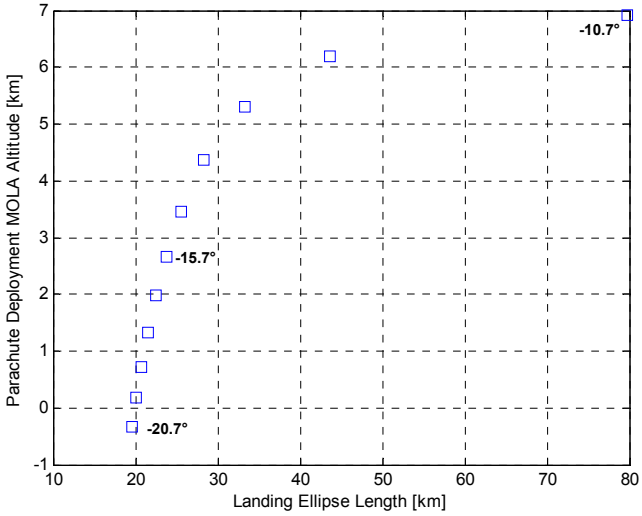


Figure 28: Parachute Deployment MOLA Altitude vs. Landing Ellipse Length for Various EFPA

G. Performance of Smart Divert for Random Terrain

For future missions that have not been defined, the exact layout of the terrain is unknown. Consequently, the quantity and arrangement of the target landing sites are additional unknowns. A Monte Carlo was performed that included the target landing site locations as uncertain parameters. The target locations were chosen randomly from the target ellipse inside the unguided parachute deployment footprint as shown in Figure 29. A 10,000 case Monte Carlo (to obtain smooth tails of the distributions) was performed for various numbers of targets, and the resulting PMF distributions may be seen in Figure 30. As expected, the required PMF necessary to divert decreases as the number of random target locations increases. Additionally, four random sites result in a required PMF less than 0.3 for 97% of the cases assuming MSL state-of-the-art interplanetary

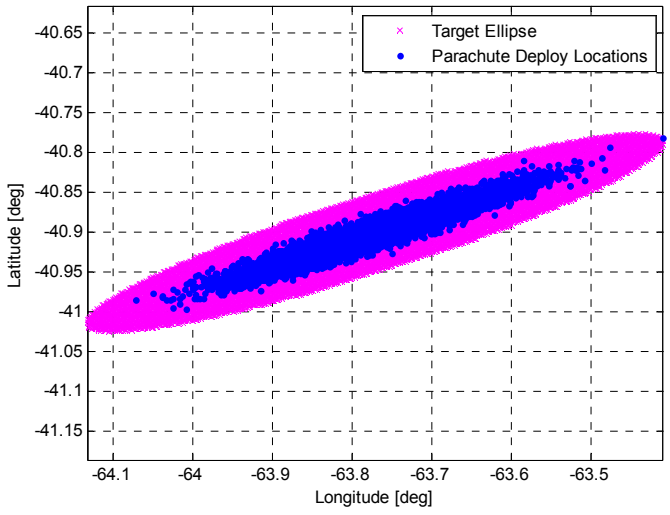


Figure 29: Target Ellipse Fitted to Parachute Deploy Locations

navigation. It is important to note that a tail of the distribution does not exist on the low PMF values. This is due to the choice of selecting the random sites inside the unguided parachute deployment footprint. The lower bound on PMF for each case corresponds to the PMF required for the trajectory at the toe of parachute deployment footprint to divert to the toe of the random site ellipse. In order to reduce the PMF for trajectories near the toe, sites should be selected farther downrange. This could be performed by shifting the target ellipse downrange by a specified bias.

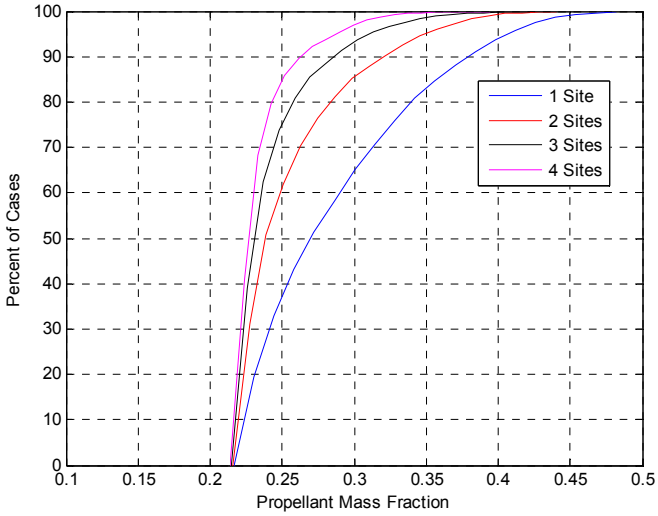


Figure 30: Cumulative Distribution Function of Propellant Mass Fraction for Various Numbers of Sites

H. Optimal Landing Site Arrangement

In reality, the safe zones will not be randomly placed inside the landing ellipse. Instead, mission designers will have the opportunity to arrange the safe zones in order to minimize required propellant. In order to reduce execution time, only 100 dispersed cases were used in the Monte Carlo. Similar to the random terrain analysis, the potential safe zones were chosen underneath the parachute deployment footprint resulting from the 100 case Monte Carlo. The safe zones were discretized inside target ellipse that was fitting around parachute deploy locations, see Figure 31. The optimal arrangement for various numbers of the safe zones was obtained to minimize the maximum required PMF. The optimization was performed using a grid search. This was performed to reduce computational expense when analyzing the optimal arrangement of various numbers of sites. For example, when identifying the location of one optimally placed safe zone, many iterations would have to be performed. Each iteration would require a Monte Carlo of 100 dispersed cases to be executed. This process would have to be repeated when optimizing the arrangement of two, three, or more safe zones and would result in redundant generated data as each optimization evaluates the same region of sites.

Instead of optimizing each individual case, a grid of 485 equally spaced safe zones was assessed. The

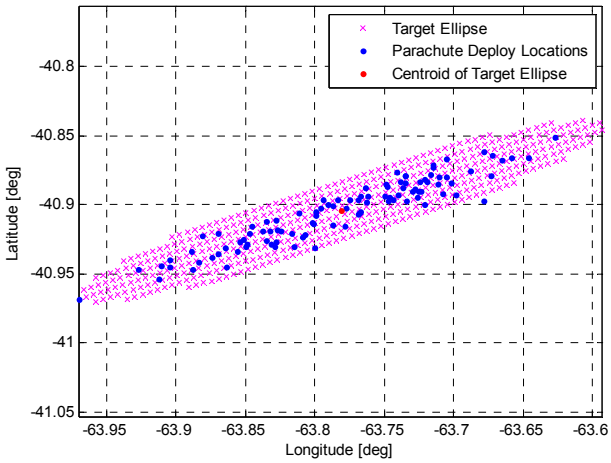


Figure 31: Potential Smart Divert Sites Fitted Around Parachute Deploy Locations

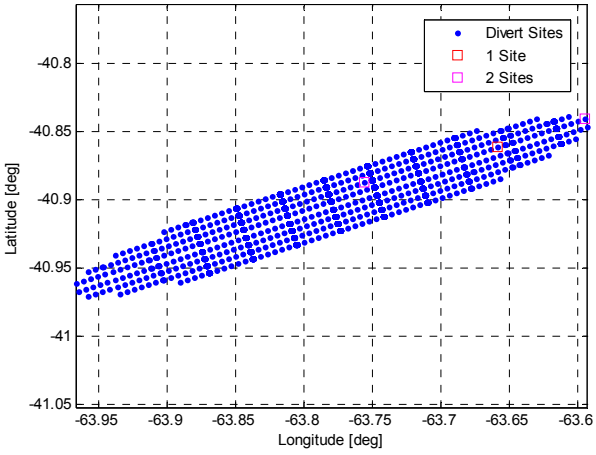


Figure 32: Divert Site Optimization

Monte Carlo of 100 dispersed cases was run for each safe zone. The 48,500 entry simulations were performed in less than 10 hours on a single computer, capitalizing on the C autocoding in Matlab and provided the data necessary to optimally arrange any number of safe zones quickly. The optimal site arrangement for various numbers of safe zones may be seen in Figure 32. As expected, the optimal arrangement consisted of divert sites located along the centerline of the target ellipse. The resulting maximum required PMF for the various number of sites may be seen in Table 19. As can be seen, two optimally arrange safe zones result in one of the safe zones placed at the toe of the ballistic parachute deployment footprint. This is due to the construction of the available safe zones directly underneath the ballistic parachute deployment footprint. Hence, the trajectory at the toe of the footprint is limiting the ability to further reduce the maximum required PMF below 0.2074, similar to that shown for the random terrain.

Table 19: Maximum Required PMF for Various Number of Safe Zones

Number of Safe Zones	Maximum Required PMF
1	0.2545
2	0.2074
3 or more	0.2074

It is clear that further analysis must be performed to understand the optimal amount of downrange bias that must be included in the placement of the target ellipse to minimize PMF. The small required PMF is also due to the rather small footprint of only approximately 20 km in major axis length originating from the 100 dispersed cases chosen for this analysis. Further analysis should include a preselected 100 cases that are chosen throughout the full 10,000 case Monte Carlo parachute deploy ellipse to properly stress the propellant required to divert to each candidate divert site during the grid search.

I. Phoenix Example

In order to demonstrate the capability of Smart Divert for a real mission scenario, rock count data for the Phoenix landing region was obtained. A contour of the rock count data may be seen in Figure 33.[7] The red regions correspond to approximately 250 observed rocks per hectare and the dark blue regions correspond to very few observed rocks per hectare. The rock count data is constructed from observation from orbit using the High Resolution Imaging Science Experiment (HiRISE) located on the Mars Reconnaissance Orbiter (MRO). The resolution of HiRISE allows the identification of rocks 1.5 m in diameter or larger. These rocks are counted by hand and by using a computerized auto-rock counter. The rock counting is performed by identifying shadows cast by rocks and large changes in albedo caused by dust surrounding rocks. As can be seen in Figure 33, various options exist to orient the landing ellipse for Phoenix (denoted by the white and gold ellipses). It is clear that mission designers are forced to place the ellipses over regions with fewer rocks to maximize the probability of landing safely. However, not all regions of the various landing ellipse are safe with certain portions of the ellipses crossing regions with high rock counts. This is likely unavoidable due to the large landing ellipse major axis lengths of approximately 200 km caused by poor interplanetary navigation

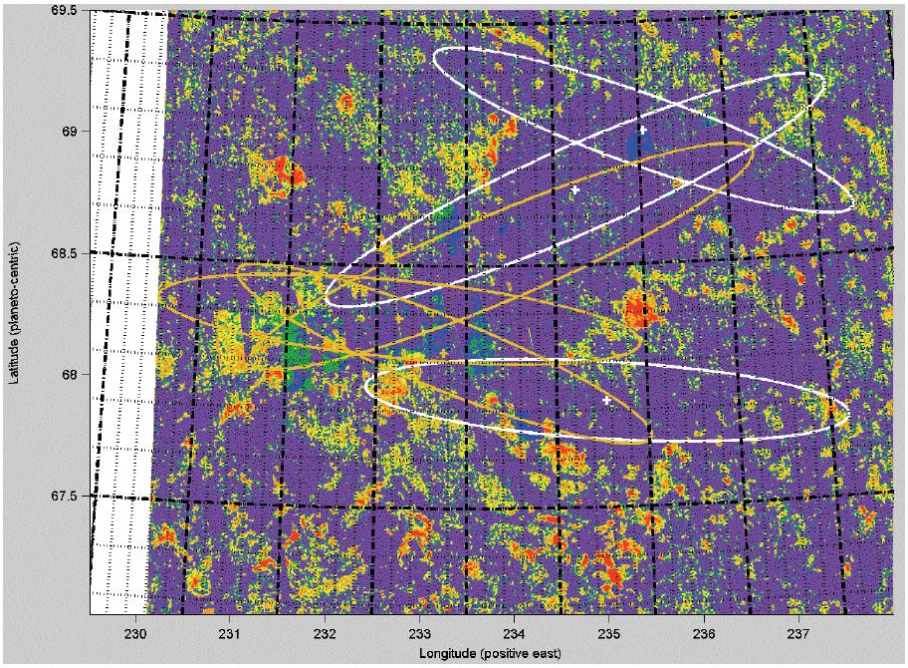


Figure 33: Phoenix Landing Region Rock Count

accuracy to reduce cost. Phoenix does not spin during the interplanetary transfer and uses a reaction control system (RCS) to maintain attitude, eliminating the need for a cruise stage. DDOR is used for Phoenix. However, the interplanetary geometry is not as good as the MER missions for DDOR. Hence, these measurements, although still useful, do not provide as high of navigational accuracy. This, in conjunction with cost saving measures, results in a rather large uncertainty in Phoenix delivery accuracy to Mars.

If the vehicle was capable of autonomously diverting to one of several apriori identified fuel-optimal safe zones, then landing ellipses could be placed over regions with many more rocks. Thus, Smart Divert may provide a way to land in these previously unreachable dangerous regions. A subset of the Phoenix region is shown in Figure 36. Note that the scaling of colors has changed with the red regions now corresponding to 50 rocks 1.5 m in diameter or larger per hectare. This would still be considered an extremely dangerous region to land. No previous lander, Phoenix, or MSL would be capable of safely landing in this region. However, this region might be very scientifically interesting due to rock abundance. Blue safe zone regions, denoted by magenta stars, with few rocks can be found embedded with the dangerous red regions with many rocks. The geometry of the magenta safe zones at the Phoenix landing region altitude of -4 km MOLA were preserved and placed under the ballistic parachute deploy footprint resulting from the MSL covariance, see Figure 34. At Mach 0.8, the parachute is released and the vehicle propulsively diverts to the fuel-optimal safe zone. The resulting cumulative distribution function of PMF may be seen in Figure 35. For this mission, Smart Divert requires a PMF less than 0.2 for all cases. Hence, Smart Divert is a simple EDL architecture capable of safely landing a vehicle in hazardous, scientifically interesting terrain. It is important to note that favorable PMF values were obtained by biasing the safe zones downrange from the parachute deploy ellipse as expected based on the random terrain analysis and landing site arrangement optimization.

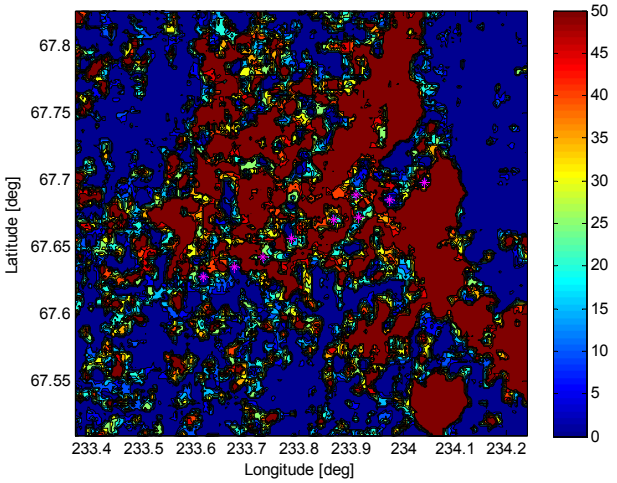


Figure 36: Subset of Phoenix Landing Site Rock Count Data per Hectare

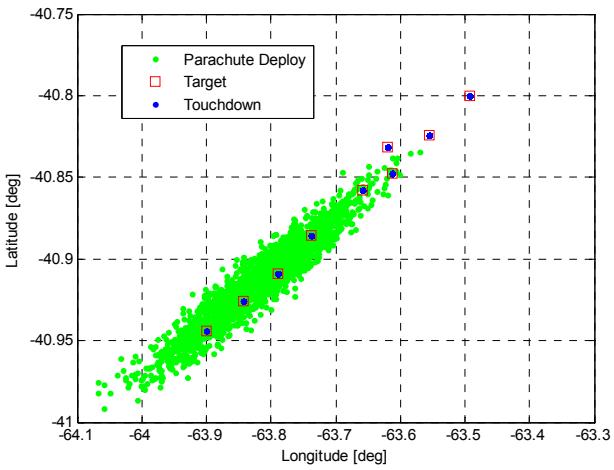


Figure 34: Snapshots of Various States

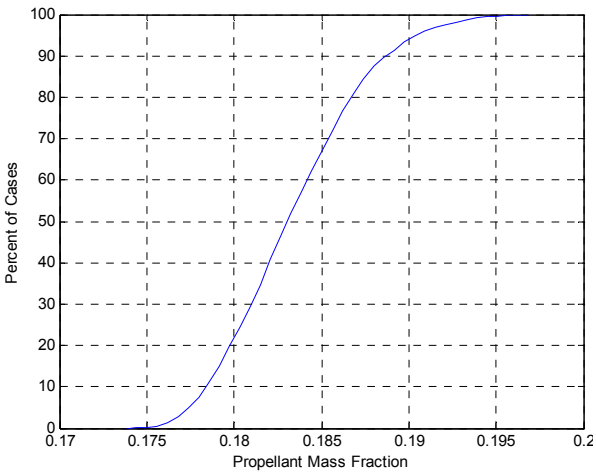


Figure 35: Cumulative Distribution Function of Propellant Mass Fraction

J. Simulator Validation

A Pathfinder test case was used to validate the developed simulation. For Pathfinder, the 585 kg vehicle entered ballistically and deployed a 12.5 m diameter supersonic disk-gap-band (DGB) parachute at a dynamic pressure of 585 Pa. At a time of 20 seconds after parachute deployment, the 64.4 kg heatshield was released. The trajectory was then propagated to the MOLA altitude immediately prior to retrorocket ignition, where the simulation was terminated. The Pathfinder entry was simulated using both the Program to Optimize Simulated Trajectories (POST) and the simulation that has been developed for this study. Figure 37 depicts both the full entry trajectory and the final phases of flight. Additionally, Table 20 compares specific trajectory event data between the two simulations. As can be seen, excellent agreement exists between both simulations.

Table 20: - Comparison of trajectory event data for Pathfinder entry

Event	POST	Simulation	% Difference
Entry			
Time (s)	0	0	0.00
Altitude (m)	128000	128000	0.00
Relative Velocity (m/s)	7479	7479	0.00
Relative Flight Path Angle (°)	-13.65	-13.65	0.00
Parachute Deploy			
Time (s)	154.5	154.3	-0.13
Altitude (m)	9916	9923	0.07
Relative Velocity (m/s)	414.5	415.2	0.17
Relative Flight Path Angle (°)	-23.35	-23.31	-0.17
Dynamic Pressure (Pa)	585.0	586.2	0.21
Heatshield Jettison			
Time (s)	174.5	174.3	-0.11
Altitude (m)	8219	8237	0.22
Relative Velocity (m/s)	90.23	90.16	-0.08
Relative Flight Path Angle (°)	-47.33	-46.56	-1.63
Dynamic Pressure (Pa)	31.98	31.88	-0.31
Trajectory Termination			
Time (s)	359.8	360.2	0.11
Altitude (m)	-2408	-2403	-0.21
Relative Velocity (m/s)	42.64	42.66	0.05
Relative Flight Path Angle (°)	-89.88	-88.83	-1.17
Dynamic Pressure (Pa)	21.55	21.55	0.00

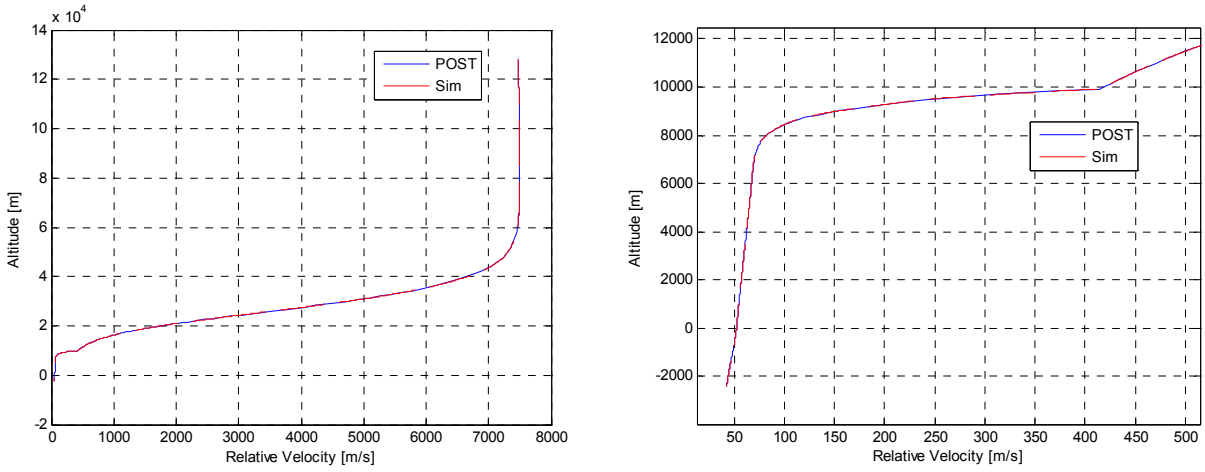


Figure 37: Comparison of Developed Simulation and POST for Pathfinder Entry

K. Conclusion

Smart Divert may enable a new, low cost EDL architecture consisting of a ballistic entry, supersonic parachute deployment, and an autonomous landing site selection. Smart Divert may also provide a simple approach to provide safe landing of vehicles in hazardous terrain. An MSL-class vehicle was studied in

order to demonstrate the capability of Smart Divert for future large Mars landers. In order to restrict the required PMF to less than 0.3 a maximum divert of 10 km or less should be initiated at 5 km AGL to provide sufficient timeline for the remaining EDL events. The altitude of the landing region governs the design of the nominal ballistic entry flight path angle. Shallower entries provide higher supersonic parachute deployment altitudes, allowing diverts to high elevations. However, shallow entries result in large landing ellipse lengths, requiring more Smart Divert safe zones to maintain reasonable PMF requirements.

The influence of number of Smart Divert sites was quantified for a random terrain in which the Smart Divert sites were randomly varied in the Monte Carlo. Four sites randomly arranged resulted in 97% of the cases requiring a PMF less than 0.3, a limit approximately double that of a gravity turn. An example method of optimal landing site arrangement showed the downrange bias necessary to optimize the maximum required PMF for one and two sites. As expected, the sites were located on the major axis of the safe zone ellipse. A downrange shift in the safe zone grid will allow optimization for more than two sites.

An example EDL scenario using rock count data from the Phoenix landing region demonstrated that Smart Divert can provide the capability to safely land entry vehicles in hazardous terrain with only a small fraction of the terrain regarded as safe. For the shown example, a minimal PMF (less than 0.2) is required to successfully perform the necessary diverts to ensure the vehicle lands safely in the hazardous terrain. Even with the introduction of hypersonic guidance for MSL, a high probability of a safe landing for this Phoenix example site could not be achieved. Hence, Smart Divert could provide the means to send vehicles to hazardous, rock populated landing areas using a simple ballistic entry followed by supersonic parachute deployment and a small divert that only minimally increases the amount of terminal descent propellant required for EDL.

L. Work That Will be Completed by Conference (Not Finished for This Draft)

It is clear that safe zones must be biased significantly downrange from the parachute deploy point to minimize the required PMF. The Smart Divert performance assessment should be expanded to incorporate flight path angle at parachute deploy to gain an understanding of what downrange bias should be used. This downrange bias should be incorporated into the random terrain analysis and landing site arrangement optimization to eliminate the trajectory at the toe of the footprint as the limiting case. This would also result in improved PMF for the random terrain analysis in which the lower PMF tail would be identified.

The influence of interplanetary navigation must also be assessed. This study assumed state-of-the-art navigation quantified in the MSL entry covariance. It is important to understand the capability of Smart Divert for cost saving, poor interplanetary navigation like that of Phoenix, resulting in large landing ellipses. Additionally, for such large landing ellipses in relatively safe terrain like that of Phoenix, the concept of Smart Divert may provide additional landing safety. Such large landing ellipse regions may contain few hazards, such as craters or large rock densities. Smart Divert may be a useful method to divert away from these dangerous exclusion zones that may sparsely exist in the large landing ellipse.

Finally, an assessment of methods to identify the fuel-optimal divert site should be performed. This analysis required an onboard evaluation of all sites by propagating simple equations of motion using the D'Souza guidance to identify the fuel-optimal safe zone. Due to the simplicity of ballistic entries, measurable entry characteristics such as peak deceleration load could potentially be used to infer the location of the vehicle inside the landing ellipse and thus can be used to select the fuel-optimal divert site without requiring the onboard propagation of equations of motion. These pre-selected sites along with measurable entry characteristics would be evaluated on the ground, eliminating the need for intelligent autonomous site selection.

M. References

[1] Braun, R.D. and Manning, R.M., "Mars Exploration Entry, Descent, and Landing Challenges," *Journal of Spacecraft and Rockets*, Vol. 44, No. 2, pp. 310-323, 2007.

[2] Mendeck, G.F. and Carman, G.L., "Guidance Design for Mars Smart Landers Using The Entry Terminal Point Controller," AIAA-2002-4502, AIAA Atmospheric Flight Mechanics Conference and Exhibit, 5-8 August 2002, Monterey, California

[3] Striepe, S., Way D., Dwyer, A., "Mars Science Laboratory Simulations for Entry, Descent, and Landing," Journal of Spacecraft and Rockets, Vol. 43, No.2, 2006, pp. 311-323.

[4] Steinfeldt, B.A., Grant, M.J., Matz, D.M., and Braun, R.D., "Guidance, Navigation, and Control Technology System Trades for Mars Pinpoint Landing," AIAA Atmospheric Flight Mechanics Conference, 18-21 August 2008, Honolulu, Hawaii (to be published)

[5] D'Souza, C., "An Optimal Guidance Law for Planetary Landing." Paper No. AIAA 97-3709, 1997.

[6] Wolf, A., Tooley, J., Ploen, S., Gromov, K., Ivanov, M., and Acikmese, B., "Performance Trades for Mars Pinpoint Landing," *2006 IEEE Aerospace Conference*, Paper 1661, Big Sky, Montana, March 2006.

[7] Spencer, D., "Landing Site Downselection," JPL Phoenix Mission Presentation.